



**Contribution à la conception, la réalisation et
l'utilisation du système de bases de données S O M I N
E. Les accès aux bases. Aide à la conception assistée par
ordinateur**

Claudette Sayettat

► **To cite this version:**

Claudette Sayettat. Contribution à la conception, la réalisation et l'utilisation du système de bases de données S O M I N E. Les accès aux bases. Aide à la conception assistée par ordinateur. Génie logiciel [cs.SE]. Ecole Nationale Supérieure des Mines de Saint-Etienne; Institut National Polytechnique de Grenoble - INPG, 1976. Français. NNT : . tel-00816088

HAL Id: tel-00816088

<https://theses.hal.science/tel-00816088>

Submitted on 19 Apr 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° d'ordre : 41S

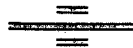
THESE

présentée par

Claudette SAYETTAT

pour obtenir

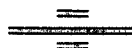
LE TITRE DE DOCTEUR DE 3° CYCLE
SPECIALITE SYSTEMES ET RESEAUX INFORMATIQUES



CONTRIBUTION A LA CONCEPTION, LA REALISATION
ET L'UTILISATION DU SYSTEME DE BASES DE DONNEES

S O M M A I R E

LES ACCES AUX BASES
AIDE A LA CONCEPTION ASSISTEE PAR ORDINATEUR



Soutenue à Saint-Etienne le 26 Janvier 1976, devant la Commission d'Examen

MM. L. BOLLIET

Président

J. BAHUAUD

R. BOUCHE

J.M. BRUN

S. GUIBOUD-RIBAUD

R. MAHL

Examineurs

N° d'ordre : 41S

THESE

présentée par

Claudette SAYETTAT

pour obtenir

LE TITRE DE DOCTEUR DE 3° CYCLE

SPECIALITE SYSTEMES ET RESEAUX INFORMATIQUES



CONTRIBUTION A LA CONCEPTION, LA REALISATION
ET L'UTILISATION DU SYSTEME DE BASES DE DONNEES

S O M M A I R E

LES ACCES AUX BASES

AIDE A LA CONCEPTION ASSISTEE PAR ORDINATEUR



Soutenue à Saint-Etienne le 26 Janvier 1976, devant la Commission d'Examen

MM. L. BOLLIET

Président

J. BAHUAUD

R. BOUCHE

J.M. BRUN

S. GUIBOUD-RIBAUD

R. MAHL

Examineurs

ECOLE NATIONALE SUPERIEURE DES MINES DE SAINT.ETIENNE.

Directeur : M. Lucien VIELLEDENT

Sous-Directeur : M. Jacques BOISSE

PROFESSEURS DE 1ère CATEGORIE

MM. COINDE Alexandre
GOUX Claude
LEVY Jacques
RIEU Jean
SOUSTELLE Michel

Gestion
Métallurgie
Métallurgie
Mécanique-Résistance des Matériaux
Chimie

PROFESSEUR ASSOCIE DE 1ère CATEGORIE

M. FORMERY Philippe

Mathématiques Appliquées

PROFESSEURS DE 2ème CATEGORIE

MM. GUIBOUD-RIBAUD Serge
LOWYS Jean Pierre
TOUCHARD Bernard

Informatique
Physique
Physique Industrielle

PROFESSEURS ASSOCIES DE 2ème CATEGORIE

MM. FONTEILLES Michel
SEROR Denis

Géologie
Informatique

DIRECTEUR DE RECHERCHE

M. LESBATS Pierre

Métallurgie

MAITRES DE RECHERCHE

MM. BISCONDI Michel
BOOS Jean-Yves
Mlle FOURDEUX Angéline
MM. LALAUZE René
LANCELOT Francis
LE COZE Jean
THEVENOT François
TRAN MINH Canh

Métallurgie
Métallurgie
Métallurgie
Chimie
Chimie
Métallurgie
Chimie
Chimie

o
o o
o

INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

Président : M Louis NEEL

Vice-Présidents : MM. Jean BENOIT
Lucien BONNETAIN

PROFESSEURS TITULAIRES

MM. BENOIT Jean	Radioélectricité
BESSON Jean	Electrochimie
BLOCH Daniel	Physique du solide
BONNETAIN Lucien	Chimie Minérale
BONNIER Etienne	Electrochimie et Electrometallurgie
BRISSENEAU Pierre	Physique du solide
BUYLE-BODIN Maurice	Electronique
COUMES André	Radioélectricité
FELICI Noël	Electrostatique
LESPINARD Georges	Mécanique
MOREAU René	Mécanique
PARIAUD Jean-Charles	Chimie-Physique
PAUTHENET René	Physique du solide
PERRET René	Servomécanisme
POLOUJADOFF Michel	Electrotechnique
SILBER Robert	Mécanique des Fluides

PROFESSEURS ASSOCIES

MM. RABINS Michaël	Automatique
ROUXEL Roland	Automatique

PROFESSEURS SANS CHAIRE

MM. BLIMAN Samuel	Electronique
COHEN Joseph	Electrotechnique
DURAND Francis	Métallurgie
FOULARD Claude	Automatique
LANCIA Roland	Electronique
VEILLON Gérard	Informatique fondamentale et appliquée
ZADWORYN François	Electronique

MAITRES DE CONFERENCES

MM. BOUDOURIS Georges	Radioélectricité
BOUVARD Maurice	Génie mécanique
CHARTIER Germain	Electronique
GUYOT Pierre	Chimie Minérale
IVANES Marcel	Electrotechnique
JOUBERT Jean-Claude	Physique du solide
LACOME Jean Louis	Géophysique
MORET Roger	Electrotechnique Nucléaire
ROBERT François	Analyse numérique
SABONNADIÈRE Jean-Claude	Informatique fondamentale et appliquée
Mme. SAUCIER Gabrièle	Informatique fondamentale et appliquée

CHARGE DE FONCTIONS DE MAITRE DE CONFERENCES

MM. ANCEAU François	Mathématiques Appliquées
PIERRARD Jean-Marie	Hydraulique.

CHERCHEURS DU C.N.R.S.

MM. FRUCHARD Robert	Directeur de recherche
ANSARA Ibrahim	Maître de recherche
DRIOLE Jean	Maître de recherche
MATHIEU Jean-Claude	Maître de recherche
MUNIER Jacques	Maître de recherche

A MON PERE

Je remercie,

Monsieur VIELLEDENT, Directeur de l'Ecole Nationale Supérieure des Mines de Saint-Etienne, qui a bien voulu m'accueillir dans son établissement pour effectuer le travail exposé dans ce mémoire.

Monsieur L. BOLLINET, Professeur à l'Université Scientifique et Médicale de Grenoble qui m'a fait l'honneur de présider ce jury

Monsieur R. MAHL, qui a guidé le démarrage de cette étude, alors qu'il était chef du département Informatique de l'E.M.S.E.

Monsieur S. GUITOUD-RIBAUD, Professeur, Chef du Département Informatique de l'Ecole qui a permis à ce projet d'aboutir.

Monsieur R. BOUCHÉ, Professeur à l'Université de LYON I, qui a dirigé l'ensemble de ce travail et auprès de qui j'ai toujours trouvé aide efficace et encouragements.

Monsieur J.M. BRUN, chargé de recherche au C.N.R.S. et Monsieur J. BAHUAUD, directeur du génie mécanique de construction de l'I.N.S.A. à LYON, pour l'aide et les conseils qu'ils m'ont apportés pour la réalisation et la rédaction de la partie concernant l'aide à la conception assistée par ordinateur.

Monsieur R. THOMAS, Chef du Centre de Calcul de l'E.M.S.E., qui m'a toujours apporté une aide spontanée.

Tous les chercheurs, techniciens et étudiants du Département Informatique et du Centre de Calcul de l'Ecole pour leur collaboration et leur accueil sympathique.

Mes collègues de l'I.U.T. qui ont su m'aider et me supporter pendant la durée de ces travaux.

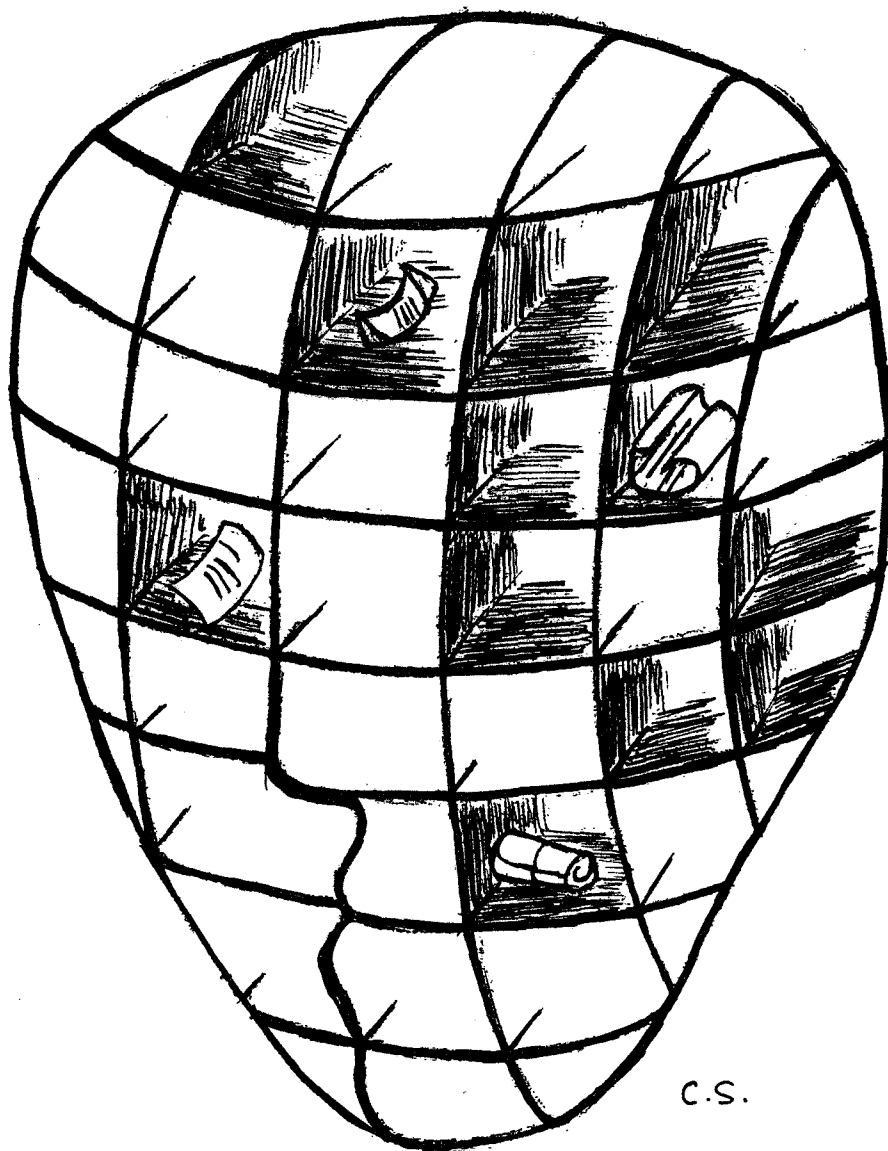
.../...

Tous ceux qui ont contribué à la réalisation matérielle de ce document, et principalement :

Mademoiselle J. DEFOUR de l'I.U.T.

Mesdames G. BONNEFOY, C. CHAMBON, M.C. MONTMARTIN, Messieurs LOUBET, BROSSARD de l'E.M.S.E.

Enfin mes deux complices, M. GAILLARD et P. MARTY avec lesquels j'ai réalisé ce travail dans l'amitié et la bonne humeur.



C.S.

" un dessin vaut mieux que cent paroles "

proverbe chinois.

SOMMAIRE DE LA THESE

	Préface
Partie I	Sommaire et présentation générale du projet
Partie II	Etat actuel du projet SOMINE Résumés sur + la gestion de mémoire + la structure des informations
Partie III	Les requêtes SOMINE
Partie IV	SOMINE et l'aide à la conception assistée par ordinateur des projets scientifiques et techniques.
Partie V	Conclusion
	Fiches signalétiques

PREFACE

Ce rapport résume le travail effectué pendant les deux dernières années par une équipe qui mène une recherche commune depuis trois ans. Nous sommes trois enseignants du département Génie Mécanique de l'Institut Universitaire de Technologie de Saint-Etienne. Réunis par nos travaux en informatique, nous avons été influencés dans le choix de notre sujet par notre profession commune et nos formations très diversifiées.

La réunion de l'informatique et de l'enseignement devait nous attirer à priori. Mais, tant dans notre pratique quotidienne (algorithmisation des solutions de problèmes, des raisonnements etc) que dans un travail de recherche de D.E.A. à l'Université de LyonI(2), nous avons rencontré les limites de l'apport de l'Enseignement Assisté par Ordinateur de forme tutorielle. Nous devons chercher autre chose !

Les rencontres avec Monsieur VIELLEDENT, Directeur de l'Ecole Supérieure des Mines de Saint-Etienne, très intéressé par les problèmes pédagogiques et avec Monsieur MAHL, Directeur du département informatique de cette école, ont contribué à définir les buts de notre recherche : implémenter un système de banques de données accessibles à des utilisateurs "étudiants" (aide à l'enseignement) "ingénieurs et techniciens" (aide à la conception assistée par ordinateur) ou "gestionnaires" (aide à la gestion).

Cet objectif fixé, nous avons étudié les réalisations françaises et étrangères dans ces domaines. Ainsi, une importante bibliographie a été consultée. Son analyse critique nous a conduit à préciser les grandes lignes de notre recherche.

Ces éléments sont résumés dans la première partie (introduction générale) de ce mémoire qui montre comment notre travail s'est orienté suivant

deux axes principaux :

a) La conception et l'implémentation d'un système de gestion de bases de données (SOMINE)

b) Les recherches montrant comment ce système peut être appliqué à des domaines aussi divers que l'E.A.O., la C.A.O. ou l'optimisation de la structuration des informations.

Un tel travail ne pouvait être réalisé que par une équipe et c'est bien dans ce cadre que les choix principaux ont été fixés et les diverses solutions discutées. Mais, dès le début de notre travail, nous avons eu soin de donner à chacun la responsabilité d'une partie du travail. Grâce à cela, nous présentons aujourd'hui trois thèses de DOCTORAT du TROISIEME CYCLE qui, issue d'un travail commun, sont cependant indépendantes. Elles portent sur les sujets suivants :

* Interface entre SOMINE et ses utilisateurs et aide à la Conception Assistée par ordinateur (présentée par Cl. SAYETTAT)

* Structuration des informations dans SOMINE et recherche des structures optimales (présentée par P. MARTY)

* Gestion des mémoires de SOMINE et Enseignement Assisté par Ordinateur (présentée par M. GAILLARD)

Conscients d'avoir rencontré plus de questions que nous n'en avons résolues, nous espérons apporter cependant, par ce travail, une certaine contribution à la solution des problèmes concernant les banques de données et leurs interfaces avec des applications diverses.

PARTIE 1

INTRODUCTION

SOMMAIRE DE LA PREMIERE PARTIE

(INTRODUCTION GENERALE)

	PAGE
1 : <u>LES BUTS DE NOTRE PROJET</u>	I. 3
2 : <u>DE LA "RECHERCHE DOCUMENTAIRE" aux "QUALITES SOUHAITABLES"</u>	7
2.1 - LES BANQUES DE DONNEES ET L'ENSEIGNEMENT	7
2.2 - LES BANQUES DE DONNEES ET LA CONCEPTION DES PROJETS SCIENTIFIQUES ET INDUSTRIELS	8
2.3 - LES BANQUES DE DONNEES ET LA GESTION	10
2.4 - LES QUALITES SOUHAITABLES	11
3 : <u>DE SOCRATE A SOMINE</u>	13
3.1 - QUELQUES RAPPELS SUR SOCRATE	13
3.1.1. - La gestion de mémoire	14
3.1.2. - La structuration des informations	14
3.1.3. - Le langage de requête	15
3.1.4. - Le langage de commande	16
3.1.5. - La programmation	17
3.2 - SOCRATE ET NOS EXIGENCES	17
3.2.1. - Choix de certains éléments de SOCRATE	17
3.2.2. - Les différences indispensables	18
3.3 - SOMINE	20
4 : <u>CARACTERISTIQUES GENERALES DU SYSTEME ET PLAN DU RAPPORT</u>	21
4.1 - LA PROGRAMMATION	21
4.2 - LA CONFIGURATION GENERALE DU SYSTEME	21
4.3 - VUE D'ENSEMBLE SUR LE TRAVAIL REALISE	24
4.4 - CONTENU DU PRESENT RAPPORT	27
ANNEXE : <u>BIBLIOGRAPHIE DE LA PREMIERE PARTIE</u> =====	29

1. - LES BUTS DE NOTRE PROJET

De par sa composition et son travail antérieur, notre équipe était sensibilisée aux problèmes de :

- l'enseignement assisté par ordinateur,
- l'aide à la conception et la simulation des phénomènes scientifiques ou techniques.

Les ordinateurs ont été utilisés largement pour l'un ou l'autre de ces deux thèmes, car, dans chacun d'eux, on trouve une partie importante de recherche documentaire.

La documentation peut utiliser des supports bien différents. Le critère essentiel pour qualifier ces supports est le temps d'accès à l'information. D'où l'idée, aujourd'hui banale, d'utiliser des mémoires d'ordinateurs comme support d'information. Il y a trois avantages à cela :

- les mémoires, disques ou bandes, sont vastes
- les temps d'accès sont beaucoup plus courts que sur les supports traditionnels (livres, etc...)
- on peut facilement actualiser les informations (effacement, création, mise à jour)

Documenter étant au centre de nos préoccupations, deux voies pouvaient être choisies :

- créer des systèmes spécialisés et distincts pour assurer l'enseignement, l'aide à la conception ou la simulation (jeux d'entreprises..)

- créer un seul système composé de centres documentaires importants et de sous-systèmes assurant les interfaces de ces centres avec ses divers "clients", qu'ils soient des "étudiants" désirant un certain "cours", des "techniciens" voulant comparer un projet à l'ensemble des réalisations existantes, des "gestionnaires" ou des "scientifiques" voulant utiliser un grand nombre d'informations dynamiques.

Les systèmes spécialisés existent. Les avantages et les inconvénients des systèmes spécialisés dans l'enseignement assisté par ordinateur sont connus (1). Nous avons nous-même utilisé un tel système lors de notre travail de D.E.A. à Lyon (2). Notre cours de FORTRAN, opérationnel depuis deux ans, nous a permis de connaître les opinions des élèves ; ceux-ci sont généralement satisfaits mais se plaignent parfois d'un certain manque de souplesse. Les systèmes spécialisés dans l'aide au projet (voir notamment le remarquable travail de M. LATOMBE (3)) donnent plus de responsabilité (donc de créativité) aux étudiants mais demandent en parallèle, un cours, sur la spécialité de projet auxquels ils apportant leur aide. Quant aux jeux d'entreprises et autres simulations de phénomènes, ils font toujours appel à beaucoup d'informations que leurs programmeurs ont bien du mal à stocker dans les fichiers traditionnels.

D'où l'idée de construire un système unique, comprenant notamment :

- une banque de données qui pourrait contenir toutes les informations nécessaires à chacune des applications,
- un langage de cours permettant de gérer les relations entre les étudiants et la banque,
- des programmes utilisateurs faisant appel à la base et permettant de résoudre en calcul ou simulation des problèmes scientifiques ou techniques.

A partir de ces objectifs qui devaient conduire à :

- des réalisations en E.A.O. non directif,
- des applications nécessitant l'interrogation et la mise à jour de grandes quantités d'informations en temps réel (gestion, aide au projet, aide aux calculs scientifiques et techniques, etc...) il nous appartenait de :
 - rechercher et étudier la documentation nécessaire,
 - faire des choix et en déduire les différentes parties du projet global,
 - écrire des programmes,
 - trouver les applications concrètes et les mettre en oeuvre,
 - juger les qualités de notre travail à partir de son efficacité dans ces applications.

A partir de la recherche documentaire qui précède toutes les réalisations, nous avons dégagé les idées générales et les caractéristiques essentielles de notre projet. Nous verrons de quelle manière dans le prochain chapitre.

2. - DE LA "RECHERCHE DOCUMENTAIRE" AUX ...

.. "QUALITES SOUHAITABLES"

L'utilisation des banques de données n'est pas nouvelle. Les constructeurs, eux-mêmes, ont mis au point, pour leurs machines, des logiciels spécifiques adaptés aux problèmes documentaires.

Cependant, leurs systèmes sont surtout adaptés aux problèmes de gestion (3). Ainsi, pour rechercher les qualités nécessaires à la satisfaction de nos différents objectifs, il est important de comprendre les questions qui se posent dans l'utilisation des banques de données pour :

- l'enseignement assisté par ordinateur,
- la mise au point des projets industriels ou scientifiques,
- la gestion.

2. 1 - LES BANQUES DE DONNEES ET L'ENSEIGNEMENT :

La place des banques de données dans l'enseignement est encore restreinte. Dans leur rapport, "l'enseignement assisté par l'ordinateur en Amérique du Nord" (1), KAISER et COULON, en distinguent deux utilisations principales :

a) - banques de données utilisées comme instrument de recherche de faits (FACT RETRIEVAL). On désigne par là toutes les méthodes qui se bornent à retrouver, dans la banque de données, les informations à communiquer aux étudiants. Il n'y a aucun travail sur ces informations. Cette méthode a notamment été utilisée par l'équipe de Jaime CARBONNEL à Cambridge (10). Son principal inconvénient est la rigidité : toutes les réponses, tous les textes sont "prévus par le maître" et fixés une fois pour toutes dans la banque. L'avantage par rapport à un cours d'E.A.O. tutoriel classique n'est pas évident.

b) - banques de données utilisées comme support d'instrument déductif. Il s'agit d'utiliser les données stockées dans la banque, non pas d'une manière "brute" comme ci-dessus (a), mais comme éléments de l'information à fournir à l'étudiant. Une réponse à donner, par exemple, sera déduite de la question de l'étudiant et des informations contenues dans la banque. On connaît les travaux réalisés sur cette question par R.C. SCHANK (6), et son équipe (7). On sait aussi que, malgré des résultats remarquables le problème posé par la déduction n'est pas résolu de manière satisfaisante. Ou bien la déduction est complètement programmée et on retombe dans la fixité reprochée aux systèmes, vue en (a), ou bien elle s'effectue aveuglément ce qui demande des temps de réponse trop longs pour utilisation en mode conversationnel, c'est à dire en enseignement.

Un compromis semble pouvoir être recherché. C'est dans ce sens que vont les travaux de WINOGRAD (8) : il veut guider la recherche des informations dans la banque par une stratégie tenant compte, notamment, de ce qui "est possible dans le contexte donné". Appliquée à la pédagogie, il nous semble que cette voie peut être féconde, surtout si on la rapproche de l'idée émise par CHANDRASEKARAN (et rapportée par KAISER et COULON).

Il propose de "construire une banque de données conversationnelle qui demanderait à l'étudiant si la voie qu'elle emprunte pour accéder à sa requête lui paraît bonne". Qu'importe, en effet, la rigidité des "atomes" à enseigner si on peut les parcourir dans un ordre choisi, avec possibilité de retour en arrière, de tests, de construction de problèmes à partir de données choisies par l'étudiant

2. 2 - LES BANQUES DE DONNEES ET LA CONCEPTION DES PROJETS SCIENTIFIQUES ET INDUSTRIELS :

Dans ce domaine également, il semble que les bases de données n'occupent pas la place qui pourrait être la leur.

Faire un "projet", c'est, presque toujours, réaliser un compromis entre l'abstrait et le réel, entre la rigueur des théories scientifiques et les exigences de la réalité. Un bon "projeteur" doit connaître les réalisations existant dans son domaine de travail et maîtriser les théories scientifiques nécessaires. Tout cela constitue d'importantes masses de documentations, de connaissances qu'il faut actualiser sans cesse. Il est certain que l'utilisation de bases de données adaptées à ces problèmes faciliterait

ce travail. L'exemple des "projets d'électrotechniques" étudiés par M.LATOMBE (3) est significatif. Il faut stocker :

- des matériaux dont les caractéristiques et le nombre sont sans cesse variables,
- des coefficients modifiés chaque jour par de nouvelles découvertes techniques,
- des algorithmes de calculs qu'il faut adapter à de nouvelles formes de machines ou de composants
- etc.....

Et tout cela doit être rapidement accessible et mis à jour, puis utilisable pour des calculs sur ordinateurs.

A travers ce qui précède se dessine le rôle des banques de données comme "puissants dictionnaires du passé technique et scientifique". Mais les banques de données peuvent intervenir dans la conception et la recherche de bien d'autres façons. Par exemple, on peut se servir des informations contenues dans une banque pour simuler le fonctionnement d'un prototype. Ont été simulés ainsi des mécanismes simples (11), les machines électrotechniques dans le programme "ESPACE" (3), les caractéristiques principales d'un moteur thermique à essence fonctionnant à différents régimes (10). Ce dernier exemple est particulièrement intéressant. On peut schématiquement le décrire de la manière suivante :

Dans la banque de données, on stocke les informations :

- type du moteur
- nombre de tours par minute du vilebrequin
- cylindrée
- consommation d'essence
- couple moteur
- puissance mécanique
- rendement
- etc..

La banque peut alors être utilisée de différentes manières :

a) - Un étudiant peut "manipuler" un moteur comme s'il était tout à fait réel. Par exemple, il peut tracer, en fonction de la vitesse de rotation du vilebrequin, les courbes classiques de consommation, de puissance et de rendement...

b) - Un constructeur peut consulter la banque pour connaître les principales caractéristiques de tous les moteurs existants qui donnent une puissance de 60 kw à 4000 tr/mn

c) - Un projeteur, ayant dessiné un nouveau moteur pourra, en fonction des dimensions et autres caractéristiques qu'il a imposées à son projet, connaître "où se trouvera approximativement son moteur dans la gamme des existants" Il pourra ainsi répondre partiellement à la question : "faut-il continuer l'étude ? " avant la construction d'un prototype...

d) - etc

Enfin, parlons de l'intervention des banques de données dans la création elle-même. Toute recherche commence généralement par une recherche d'idées. Les techniques de la créativité sont connues (brainstorming, listes de questions, méthodes analogiques, logiques etc). Mais, toutes les idées étant trouvées la question qui se pose est la suivante : comment en dégager la solution optimale ? En général, c'est un "exploit" que l'on réserve aux personnes "d'expérience" ! Or, une approche possible est la comparaison des différentes solutions à l'aide de critères quantitatifs ou qualitatifs soigneusement hiérarchisés (MAUFIT (13), GEMINARD (14), SARAZIN (12)). Dans cette recherche l'analyse combinatoire joue un rôle important et on comprend la place que pourrait y tenir une banque de données dont les informations seraient accessibles directement par des programmes classiques de calcul.

2. 3 - LES BANQUES DE DONNEES ET LA GESTION :

Depuis 1960, le développement des S.G.B.D. est étroitement lié au développement de l'informatique de gestion.

L'apparition des premiers systèmes de gestion de base de données IDS (Integrated Date Store), TDMS (Time-Share Data Management System) (9), IMS (Information Management System) puis de SOCRATE et DBTG-CODASYL a permis de répondre en grande partie aux besoins des utilisateurs qui désiraient éviter les lacunes des méthodes traditionnelles de l'informatique de gestion (redondance des informations, rigidité des fichiers, difficultés d'accès aux informations...)

Actuellement, nous pouvons constater un intérêt grandissant pour les systèmes de gestion de bases de données comme en témoignent d'une

part la multiplication des rencontres organisées sur ce sujet (niveaux recherche ou utilisateur), d'autre part la rapide augmentation du nombre des logiciels disponibles et des installations existantes.

A titre d'exemple citons :

- TOTAL (développé par CINCOM système) qui compte près de 800 installations dans le monde entier dont une dizaine en France.
- I.M.S. qui est utilisé en France par de nombreux groupes bancaires, des assurances, des grandes entreprises nationalisées ou privées (4)
- I.D.S. qui compte plus de 500 installations dans le monde.
- SOCRATE plus récent dont le nombre d'installations est voisin de 40 (5).
- ADABAS, FORTE, D.M.S., GIM, SYSTEME 2000 , MISFIIT

Notons que la taille trop grande des systèmes disponibles ne permet pas encore un développement dans les petites et moyennes entreprises et que certains problèmes de performance (temps d'accès ...) seront résolus lorsque le " hardware" aura rattrapé son retard sur la technologie de conception des bases de données.

2. 4 - LES QUALITES SOUHAITABLES :

A travers la "recherche documentaire" concernant les trois types d'applications envisagées, il apparaît clairement qu'un système de bases de données peut apporter des éléments nouveaux dans les recherches concernant ces trois domaines.

Mais pour cela, il est indispensable que le système de base de données utilisé possède un certain nombre de qualités principales qu'il paraît possible de résumer ainsi :

- a) - Indépendance du système par rapport au matériel informatique utilisé : cette qualité assurera la "transportabilité" du système de bases de données et permettra de comparer éventuellement des matériels.
- b) - Fiabilité et garantie du secret : ces qualités sont indispensables à tout système de bases de données.

.../...

c) - Adaptabilité du système : cette qualité est particulièrement importante. Il ne faut pas présenter un système "rigide", "non modifiable" mais, au contraire, un système qui, tout en étant complet, peut être "adapté" à des applications particulières : la modularité de la programmation et du système lui-même devront autoriser l'utilisateur à compléter, supprimer ou ajouter certaines fonctions.

d) Mesurabilité des caractéristiques du système : Ceci est nécessaire pour évaluer ses performances mais aussi, par exemple, pour mesurer les effets d'un changement de structuration ou pour procéder à certaines expériences à partir de la variation des valeurs des caractéristiques du système en vue de son optimisation future.

e) - Traitement efficace des informations : cela comporte notamment :

- * la rapidité des opérations d'interrogation et de mise à jour
- * la souplesse dans la structuration et la recherche des informations (plusieurs types de structures doivent être possibles, la recherche des informations doit être rapide et pouvoir utiliser plusieurs moyens : accès direct, critères de filtrages, etc...)
- * la variété dans le choix des moyens de mise à jour et d'accès aux informations situées dans la base : les modes conversationnel et batch sont indispensables mais il faut surtout que des programmes puissent accéder aux informations, éventuellement les transformer puis les ranger à nouveau dans la base.
- * la variété dans les centres d'informations possibles: listes de valeurs, mots, textes, numériques de tous types, chaînes de bits etc....

Cet ensemble de "qualités souhaitables" constitue une sorte de "CAHIER DES CHARGES" du système à réaliser. Il peut servir à comparer nos objectifs aux qualités présentées par les réalisations existantes.

3. - DE SOCRATE A SOMINE

A partir du "cahier des charges" défini à la fin du chapitre précédent, il convenait d'examiner les systèmes de bases de données existants pour les comparer et pour éviter la reproduction de l'un ou de l'autre de ces systèmes.

Dans le paragraphe 2.3, nous avons donné la liste des systèmes de base de données dont nous avons consulté la documentation et, dans la conclusion de ce mémoire, nous reviendrons sur la comparaison de notre "produit" avec certains de ces systèmes.

Nous voulons cependant, dès cette introduction, traiter d'une manière spéciale le SYSTEME SOCRATE (5) car le groupe dans lequel nous avons travaillé au début de notre projet (MM. MAHL et THOMAS notamment) était en relation avec l'équipe SOCRATE de Grenoble. Nous avons donc pu :

- entrer en contact avec cette équipe,
- obtenir toute la documentation sur le système SOCRATE.

De ce point de vue, on peut dire que SOMINE est un des fils de SOCRATE..... ce qui ne veut pas dire que le fils est une "copie" du père !. Nous verrons au contraire, dès ce chapitre et dans toute la suite de ce mémoire, que ces deux produits sont très différents.

Remercions cependant les concepteurs et les réalisateurs de SOCRATE qui nous ont reçus plusieurs fois à Grenoble. Il est certain que leur réalisation nous a fortement influencés dans le début de nos travaux. Examinons leur produit.

3. 1 - QUELQUES RAPPELS SUR SOCRATE :

Ce paragraphe se contente de rappeler sommairement quelques éléments importants sur le système SOCRATE. Il suppose connues les nombreuses

brochures dans lesquelles ce système a été décrit et, principalement la publication citée en (5).

3. 1. 1 - La gestion de mémoire :

Une base de données SOCRATE utilise deux espaces mémoires secondaires : un espace "tambour" pour la structure et un espace "disque" pour les informations structurées (données).

A chacun de ces espaces secondaires correspond un espace "virtuel" de grande taille (2^{31} mots) dans lequel toute information a une adresse virtuelle.

L'unité de passage de l'espace virtuel à l'espace mémoire secondaire est appelée "sous-page" (bloc de $2^n - 1$ mots consécutifs.

L'unité de passage entre les mémoires secondaires et la mémoire centrale de l'ordinateur est appelée "page" : bloc de m sous-pages consécutives.

Une information subit donc les transferts suivants :

Information \longrightarrow espace virtuel \longrightarrow espace disque \longrightarrow espace tore
 (programmeur) (adresse virtuelle) (adresse disque) (adresse en mém.centrale)

Les applications permettant les passages entre les divers espaces mémoires sont réalisées par des fonctions de réallocations programmées, utilisant les méthodes de l'adressage dispersé, de l'adressage calculé et des tables (dictionnaire de gestion de mémoire). Elles sont entièrement transparentes à l'utilisateur de la banque.

3. 1. 2 - La structuration des informations :

La structure SOCRATE de base est arborescente. Elle ordonne des caractéristiques de divers types : valeurs numériques entières, listes de valeurs, mots, textes, entités. Ces caractéristiques peuvent être "simples" ou "complexes", regroupées en "blocs" si on le désire. Les "entités" sont des "blocs à caractère répétitif".

exemple : ENTITE 100 PERSONNES
 DEBUT
 NOM MOT 6
 FIN

Cette entité permettra le stockage des noms de 100 personnes (au plus).

Le COMPACTAGE DES DONNEES est une qualité essentielle de SOCRATE car :

- les valeurs numériques, définies par leur plage de variations sont codées sur des frontières de bit,
- les mots sont codés sur des frontières d'octet,
- les textes sont codés sur des frontières de mot.

On peut ainsi gagner de la place sur les fichiers traditionnels de trois façons :

- non redondance des informations (propriété de toutes les banques d'informations)
- compactage,
- choix de la meilleure structure à partir du critère : "meilleur compactage".

Les structures SOCRATE admettent un certain nombre de "caractéristiques particulières" qui accroissent de façon importante la souplesse des mises à jour et des interrogations et diminuent les temps nécessaires à ces opérations. Citons par exemple :

- la caractéristique "REFERENCE" : elle transforme la structure arborescente classique en une structure de GRAPHE
- les caractéristiques "discriminante rapide" et "inverse" qui diminuent les temps d'accès à certaines informations.
- certains filtres : "alors", "pour" qui permettent des sélections particulières d'information.

3. 1. 3 - Le langage de requête :

C'est le langage du dialogue entre la banque et ses utilisateurs. Dans ce langage, on utilise une technique de type COBOL (ou PL1) consistant à désigner tous les niveaux de blocs pour arriver à l'objet auquel on s'intéresse. Mais la présence des entités impose la création de "FILTRES" servant à caractériser plus précisément l'objet recherché. Les principaux filtres sont :

- AYANT : filtre de base généralement utilisé dans les requêtes. Il définit une réalisation possédant une ou plusieurs caractéristiques de valeur (de type) particulière .

- les filtres utilisés à la suite du filtre "AYANT" : " UN" "DERNIER" (positionnement dans l'entité), " LUI-MEME" et "EN COURS" (positionnement sur la caractéristique dont on vient de s'occuper ou sur laquelle on travaille actuellement), UN et TOUT (nombre d'éléments auxquels on veut s'adresser) etc.

- les filtres combinant des expressions " BOOLEENNES" : "ET", "OU".

- les démonstratifs : ils permettent de caractériser l'élément par son numéro d'ordre dans l'entité ; exemple : NOM D'UNE PERSONNE X (I) fournira le nom de la 50^{ième} personne si $X(I) = 50$.

- filtre " TEL QUE" ...

Ce langage, assez complexe peut être allégé par l'utilisation de "MACROS". Un programme " MACROGENEREATEUR " transforme alors le texte allégé en un texte normal, (rédigé en langage de citation ordinaire) qui est transmis au compilateur puis à l'interpréteur de SOCRATE.

3. 1. 4. - Le langage de commande :

Il complète le langage de requête en précisant le "mode" du travail à réaliser ou certaines particularités de ce travail. Il autorise les opérations de :

MISE A JOUR : création, effacement et mise à jour

ADDITION : on peut ajouter un nouveau représentant à une classe d'entité

INTERROGATION et DENOMBREMENT : permet de connaître le nombre des réalisations effectives d'une entité

SURVEILLANCE : en traitant une certaine partie de la structure de manière "préférentielle", cette commande autorise l'allègement du langage de citation.

POUR : permet l'utilisation de "groupements de commandes" : par exemple, plusieurs interrogations seront réalisées par une seule requête (interrogation itérative).

.../...

3. 1. 5. - La programmation :

Socrate a été programmé en autocode I.B.M. à Grenoble puis une version a été redéfinie pour les ordinateurs produits par la compagnie internationale d'informatique (série IRIS).

Dans les deux cas, le système n'est pas indépendant de la machine.

3. 2 - SOCRATE ET NOS EXIGENCES :

Comment le système SOCRATE se situe - t - il par rapport aux "qualités souhaitables" résumées dans notre cahier des charges ?

Répondre à cette question c'est à la fois :

- choisir certains éléments de Socrate qui pourront être utiles pour la conception et l'implémentation de SOMINE
- Montrer en quoi SOMINE doit différer de SOCRATE de manière fondamentale.

3. 2. 1. - Choix de certains éléments de Socrate utilisables dans SOMINE :

Certaines idées originales du système SOCRATE peuvent être conservées : il en est ainsi dans la GESTION DE MEMOIRE pour :

- * l'utilisation d'un fichier-structure et un fichier-données
- * la virtualisation de la mémoire et le partage de la mémoire réelle en espace disque et espace tore.

Dans la structuration des informations on peut conserver :

- la structuration arborescente dans le cas général,
- la caractéristique "référence" dans les cas où des structures de "graphes" sont indispensables.
- l'idée des "filtres" mais en transformant considérablement leur conception et leur utilisation car les filtres SOCRATE nous paraissent trop complexes, parfois redondant et pas tous utiles.

Au niveau du langage de requête, il faut retenir :

- la désignation successive de tous les niveaux de la structure qui permettent d'arriver jusqu'à l'objet (ou le groupe d'objets-entité) auquel on s'intéresse.
- les démonstratifs qui achèvent, de façon claire, cette détermination.

.../...

- certains éléments du langage (UN, TOUT, etc) mais il est absolument indispensable de simplifier et d'élaguer ce langage de requête si on veut le rendre facilement utilisable par des programmes.

Enfin, les modes "création", "interrogation", "mise à jour" "suppression", sont indispensables tandis que le "dénombrement" des réalisations effectivement créées est seulement utile.

La simplification du langage de requête devrait rendre sans objet les modes "surveillance" et "pour".

Au fil de cette analyse, on voit qu'il faut surtout transformer les parties qui sont plus ou moins directement en relation avec les utilisateurs. La comparaison des qualités de Socrate avec les qualités que nous avons estimées importantes confirme ce point de vue.

3. 2. 2. - Les différences indispensables :

Reprenons nos critères un à un en insistant maintenant sur les aspects nouveaux que SOMINE doit présenter par rapport à SOCRATE.

3.2.2.I - Indépendance du système par rapport au matériel informatique :

Les réalisations actuelles du système SOCRATE ne sont pas transportables. Ainsi, une adaptation à un nouveau matériel impose une nouvelle programmation des algorithmes. SOMINE, au contraire, DOIT ETRE TRANSPORTABLE. Pour cela, nous devons programmer dans un langage algorithmique.

3.2.2.II - Fiabilité et garantie du secret :

SOCRATE est fiable et le secret des informations y est correctement assuré. SOMINE n'apportera pas de nouveautés sur ce point.

3.2.2.III - Adaptabilité du système :

SOCRATE est un système assez vaste et rapide. Ces qualités ont pour revers des difficultés d'adaptation en vue d'une application particulière. Sa programmation en langage autocode ne permet pas à l'utilisateur moyen de comprendre son fonctionnement.

La programmation de SOMINE devra être la plus COMPREHENSIBLE possible (choix du FORTRAN, modularité). Sur ce point SOMINE et SOCRATE seront donc très différents.

.../...

3.2.2.IV - Mesurabilité des caractéristiques du système :

Toutes les caractéristiques de SOCRATE (du nombre de pages en mémoire centrale au temps de réponse dans telle condition de fonctionnement) ont été mesurées. Il reste cependant un certain nombre d'expériences à réaliser. Nous pensons notamment à des expérimentations sur la taille des pages, sur l'effet de la présence ou de l'absence du cadre de manoeuvre, sur le nombre optimum de pages à garder simultanément en mémoire centrale.... etc. Les transformations de programmes nécessaires pour ces expériences ne sont pas faciles sur le système SOCRATE. Il faudra que SOMINE les permettent. D'autre part, il faudrait pouvoir compter les nombres d'interrogations et de mises à jour afin d'effectuer des recherches statistiques sur le fonctionnement de la base.

3.2.2.V - Traitement efficace des informations :

- a) Le temps de réponse de SOCRATE est satisfaisant. SOMINE, dans sa version originale essaiera seulement de... faire aussi bien.
- b) Les divers modes de structuration des informations ainsi que la variété des moyens d'accès sont un des atouts de SOCRATE. Par contre, la complexité du langage de requête est un défaut; il est souhaitable que l'utilisateur de la banque dispose d'un langage plus clair, plus proche du langage naturel. SOMINE s'efforcera de joindre cette qualité à une rapidité d'accès convenable.
- c) SOCRATE peut être utilisé à partir de télétypes en mode conversationnel. Mais, les programmes ne peuvent pas utiliser directement la banque et la banque ne peut pas lancer l'exécution d'un programme. Au contraire, pour atteindre ces objectifs (notamment pour le calcul scientifique) il est INDISPENSABLE QUE LE SYSTEME SOMINE PUISSE A LA FOIS :

- être accessible par tout programme en langage évolué
- provoquer l'exécution de programmes.

A partir de ce critère apparaît une des différences essentielles entre les deux systèmes.

En effet, l'exécution de programmes à partir de la base elle-même est une propriété très importante car, à partir d'une seule requête on pourra déclencher l'exécution d'un nombre quelconque de requêtes et de programmes. Cela ouvre la voie à l'utilisation directe d'une banque de données dans les domaines de la conception assistée par ordinateur, du calcul scientifique et technique et résoud le problème des actions automatiques sur les informations.

.../...

d) Tous les types d'informations retenus dans SOCRATE le seront également ans SOMINE (entités, listes de valeurs, textes, mots, numériques entiers). Mais les calculs souhaités (aide à la conception, par exemple) imposent d'autres types d'informations : les valeurs numériques REELLES et, surtout, les PROGRAMMES.

3.3 - SOMINE :

SOMINE est donc fils de SOCRATE.

Mais SOMINE n'est pas SOCRATE car la discussion précédente fait apparaître trois sortes de différences :

- au niveau du mode et du langage de programmation
- au niveau des types d'informations et, surtout des types d'UTILISATEURS puisque SOMINE acceptera comme utilisateurs des PROGRAMMES EN LANGAGE EVOLUE.

- au niveau des langages de définition de structure et de requête : ces langages seront simplifiés et clarifiés afin de limiter au minimum le besoin de formation des utilisateurs de SOMINE.

Ayant défini ce que "nous souhaitions réaliser" dans les deux chapitres dont la lecture s'achève ici, nous devons présenter les caractéristiques générales de SOMINE et le plan du mémoire. Ce sera l'objet du chapitre suivant.

4. - CARACTERISTIQUES GENERALES DU SYSTEME ET PLAN DU RAPPORT

Le "cahier des charges" défini à la fin du précédent chapitre nous a guidé au cours de la conception et de l'implémentation de la base de données SOMINE et de ses applications. Il nous a imposé un certain nombre de choix essentiels au niveau de :

- la programmation,
- la configuration générale du système global d'aide à l'enseignement, à la conception et à la gestion,
- la conception et l'implémentation de SOMINE,
- la mise en oeuvre des applications.

Ce sont ces choix généraux que nous allons présenter en renvoyant chaque fois au rapport dans lequel ils sont développés.

4. 1 - LA PROGRAMMATION :

Nous avons voulu réaliser un système indépendant du matériel utilisé. Pour cela, nous avons écrit tous les programmes (sauf deux sous-programmes de petite dimension) en FORTRAN.

Par souci de souplesse, la programmation a été réalisée de façon modulaire ce qui permet d'ajouter, d'enlever ou de modifier les fonctions de base du système.

4. 2 - LA CONFIGURATION GENERALE DU SYSTEME :

Le système global d'aide à l'enseignement, à la conception et à la gestion est construit autour d'un système de base de données (SOMINE) dont les "caractéristiques" très diversifiées (numérique entier et décimal),

.../...

texte, liste de valeurs, programmes d'écriture de cours ou programmes de tous types compatibles avec FORTRAN) permettent des applications multiples.

Conformément à la qualité principale souhaitée, ce système de bases de données peut être utilisé directement par des personnes (mode conversationnel sur écrans (16) alphanumériques et graphiques) ou par des programmes rédigés en langages compatibles avec FORTRAN.

Un "programme-tampon" de temps partagé assure l'interface entre les utilisateurs travaillant "simultanément" et le système de base de données, conformément à la figure de la page suivante.

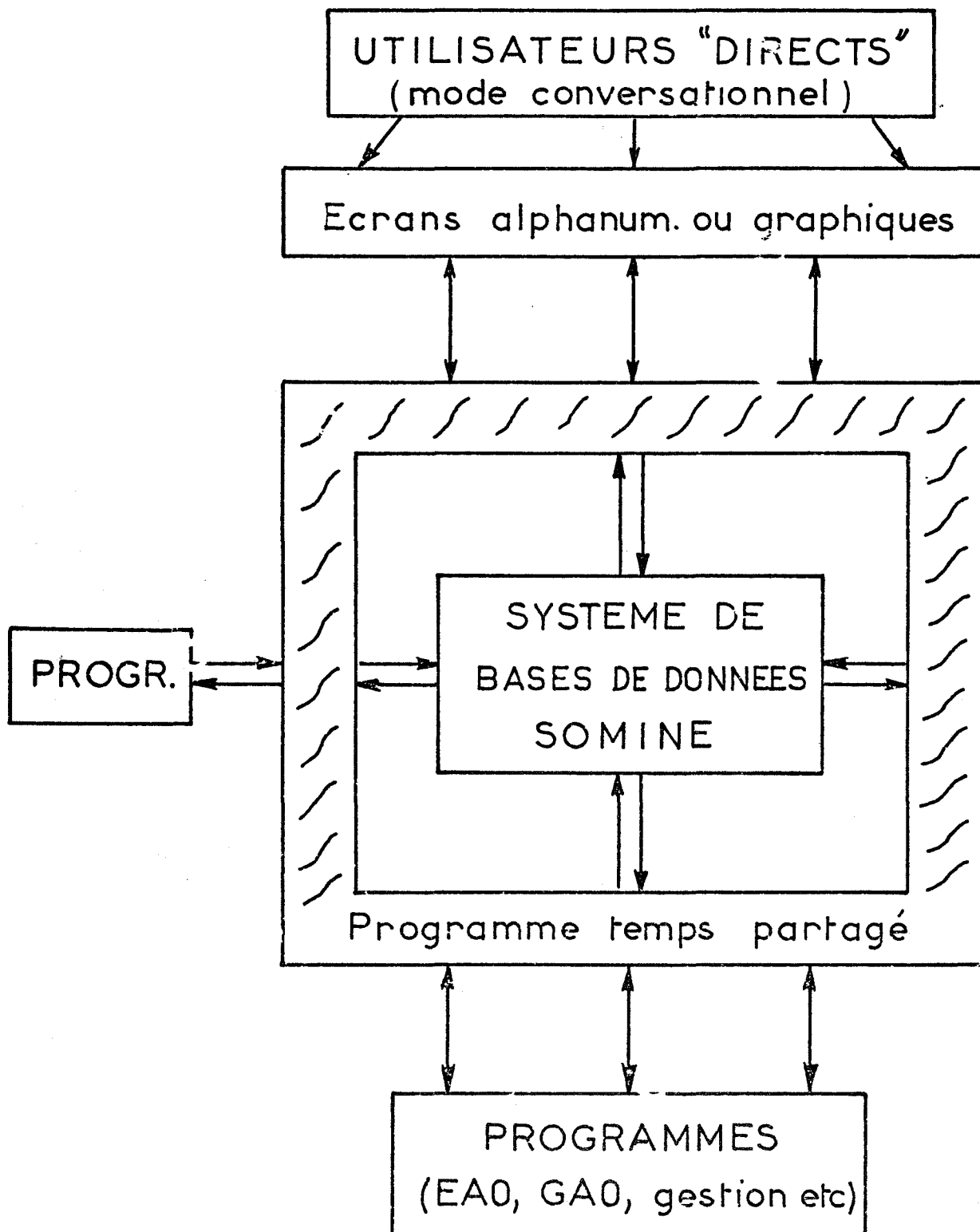


Figure I.1 : SOMINE CENTRE D'UN SYSTEME

4. 3 - VUE D'ENSEMBLE SUR LE TRAVAIL REALISE :

L'implémentation d'un système de gestion de bases de données pose des problèmes dans les domaines de :

- la gestion de la mémoire
- la structuration des informations
- l'interface entre le système et ses utilisateurs.

a) gestion des mémoires :

Une banque de données est un ensemble d'objets (informations et relations) reliés entre eux par une structure.

Ces objets ont des caractères dynamiques : leurs places et leurs tailles peuvent changer au cours d'un traitement.

La place physique nécessaire pour "loger" ces objets demande un espace important (de l'ordre de cent millions d'octets dans la plupart des applications de gestion). Il est donc impossible, dans l'état actuel de la technologie, d'implémenter un système de bases de données sans assurer la répartition des informations sur plusieurs niveaux de mémoire physique. Assurer cette répartition et les passages entre les divers espaces en donnant à l'utilisateur de la banque l'impression qu'il travaille dans un espace UNIQUE, de taille pratiquement ILLIMITEE, constituent les fonctions principales de la gestion de mémoire. La conception et la réalisation de la gestion des mémoires de SOMINE seront exposées dans la thèse de M. GAILLARD.

b) Structuration des informations :

Les performances d'un système de base de données dépendent largement des divers choix opérés lors de la définition de l'implémentation de la structure formelle des informations.

Les diverses structurations permises par SOMINE et le langage au demeurant extrêmement simple, qui permet leur implémentation seront définis et commentés dans la thèse de P. MARTY.

c) SOMINE et ses utilisateurs :

La structure des informations étant définie et implémentée, il faut maintenant "remplir" la banque, puis l'interroger. Cela concerne le

traitement des informations c'est à dire, finalement, les relations entre SOMINE et les utilisateurs. L'accès à la banque se fait à travers un langage de requête qui doit être simple et adapté à tous les types d'applications.

La conception et l'implémentation du langage de requête ainsi que les diverses possibilités de traitement des informations que SOMINE peut offrir sont exposées dans ce mémoire.

D'autre part, puisque nous étions partis d'un "cahier des charges" il était important de vérifier que le système construit possédait les propriétés qui avaient motivées sa conception. Aussi, nous avons développé trois recherches dans des domaines divers afin de mettre en évidence les qualités et les insuffisances de notre système et de montrer comment il était possible de le compléter pour pallier à ces insuffisances.

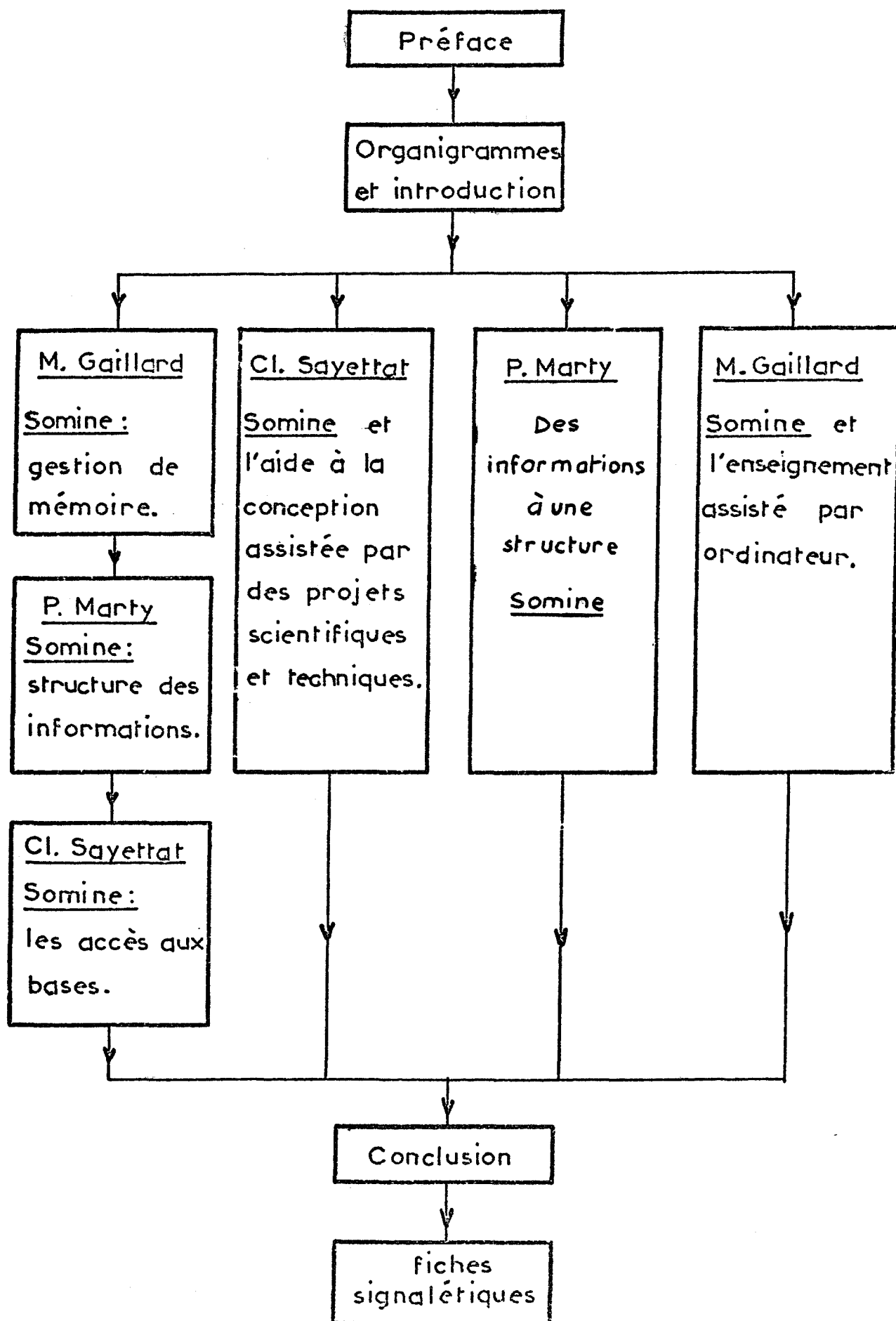
Les comptes-rendus de ces travaux sont exposés :

- dans la thèse de M. GAILLARD pour l'Enseignement Assisté par Ordinateur,
- dans la thèse de P. MARTY pour la Structuration Optimale des informations.

Dans le présent document le lecteur trouvera l'application de SOMINE à l'Aide à la conception assistée par ordinateur (partie IV).

Ces trois thèses sont le résultat d'un travail d'équipe cependant elles peuvent être lues indépendamment l'une de l'autre.

L'organigramme de l'état actuel du projet SOMINE se trouve à la page suivante.

ETAT ACTUEL DU PROJET SOMINE

4. 4 - CONTENU DU PRESENT RAPPORT :

Dans la suite de ce mémoire on trouvera :

- des résumés sur la gestion des mémoires et la structuration des informations (partie II)
- les accès aux bases SOMINE (partie III)
- SOMINE et l'aide à la conception assistée par ordinateur (partie IV)
- une conclusion générale (partie V) dans laquelle nous donnons le bilan actuel des résultats obtenus par notre équipe.

Nous espérons donner ainsi à la fois une vue exacte de notre travail et une vue d'ensemble sur le projet dans lequel notre travail est inclu.

Nous sommes à la disposition de toutes les personnes qui désireraient des compléments d'information.

PARTIE II

Ce rapport est une contribution à la conception, la réalisation et l'utilisation du système de bases de données SOMINE.

Les pages qui suivent ont pour but de faciliter la compréhension du projet SOMINE en donnant quelques éléments contenus dans :

- la gestion des mémoires (thèse de M. GAILLARD)
- structuration des informations dans les bases SOMINE (thèse de P. MARTY)

Pour plus de précisions sur les options retenues et sur la réalisation des parties du prototype SOMINE qui ne font pas l'objet de ce rapport, le lecteur pourra se reporter aux deux thèses nommées ci-dessus.

Quelques remarques sur la gestion des mémoires

La gestion des mémoires réalise les passages de l'espace réel disque à l'espace virtuel et inversement ceci à partir de la mémoire centrale de l'ordinateur. Pour cela, il y a projection d'un espace virtuel de dimension très grande sur un fichier disque de taille plus petite. La méthode retenue est :

Une gestion de mémoire paginée allouée de façon statique.

Elle réalise les fonctions suivantes :

- * Création et préformatage des fichiers disques .
- * Formatage et initialisation des fichiers disques
(programme FORMFICH)
- * Ouverture des fichiers (programme INFICH)
- * Sauvegarde des informations et fermeture des fichiers
(programme OUTFICH)
- * Relations entre mémoire centrale et fichiers disques .
- * Lecture des informations (programme LECTURE)
- * Ecriture des informations (programme ECRIRE)

Voici SOMINE

SOMINE permet de construire des bases de données : si vous avez des informations de toutes natures : mots, textes, valeurs numériques entières ou réelles..... si vous désirez les conserver, les retrouver, les modifier, les supprimer, les dénombrer, les associer,..... SOMINE peut le faire pour vous.

Les bases de données dettype SOMINE sont des bases à deux fichiers.

* le fichier structure qui contient la structure des informations à mettre dans la base.

* le fichier données qui contient les informations elles-mêmes.

Ces deux fichiers sont virtuellement de taille très grande et seules les sous-pages non vides sont transférées sur l'espace réel disque. C'est la gestion de mémoire décrite page II-3 qui réalise le passage de l'espace réel à l'espace virtuel et inversement, ces deux actions étant transparentes à l'utilisateur de SOMINE.

Lettravail de création et d'utilisation d'une base de données SOMINE comprend quatre phases :

- 1 - Rassemblement des informations qui doivent être mises dans la banque.
- 2 - Définition de la structure formelle suivant laquelle les informations seront stockées (utilisation du langage de définition de structure).
- 3 - Analyse syntaxique et compilation de la structure ainsi définie.
- 4 - Traitement des données à l'aide des requêtes mises au point, par exemple : la création, la suppression, l'interrogation , la mise à jour, la fréquence d'accès à un élément d'information.

1 - ANALYSE DU PROBLEME :

Un essai de formalisation a été réalisé dans la thèse de P. MARTY (structuration des informations).

.../...

2 - STRUCTURATION DE L'INFORMATION :

2. 1. - Structure :

L'information doit être organisée en graphe. La structure est la description linéaire de cet arbre par la méthode du préordre. Dans l'exemple ci-dessous, l'arbre est décrit dans l'ordre alphabétique.

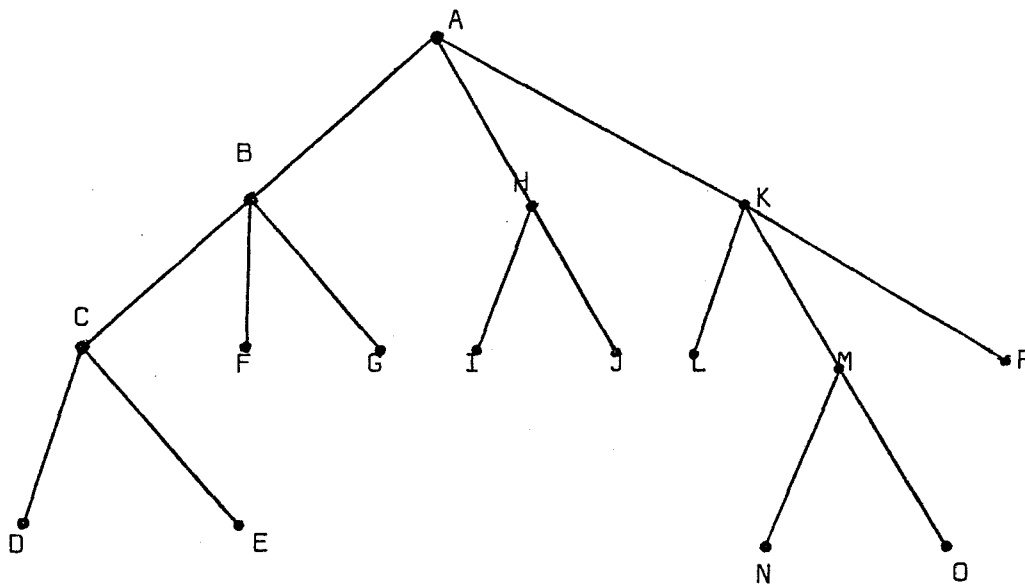


Figure II.1

Le langage de définition de structure permet cette description.

2. 2. - Langage de Définition de Structure : (LDS)

2.2.1. - Grammaire :

Les règles syntaxiques sont fournies par les automates données page 6 .

A chaque unité d'information ou caractéristique sont associés :

- un identificateur pour la désigner,
- un type qui permet de connaître sa nature,

Les automates de définition du langage de structure .

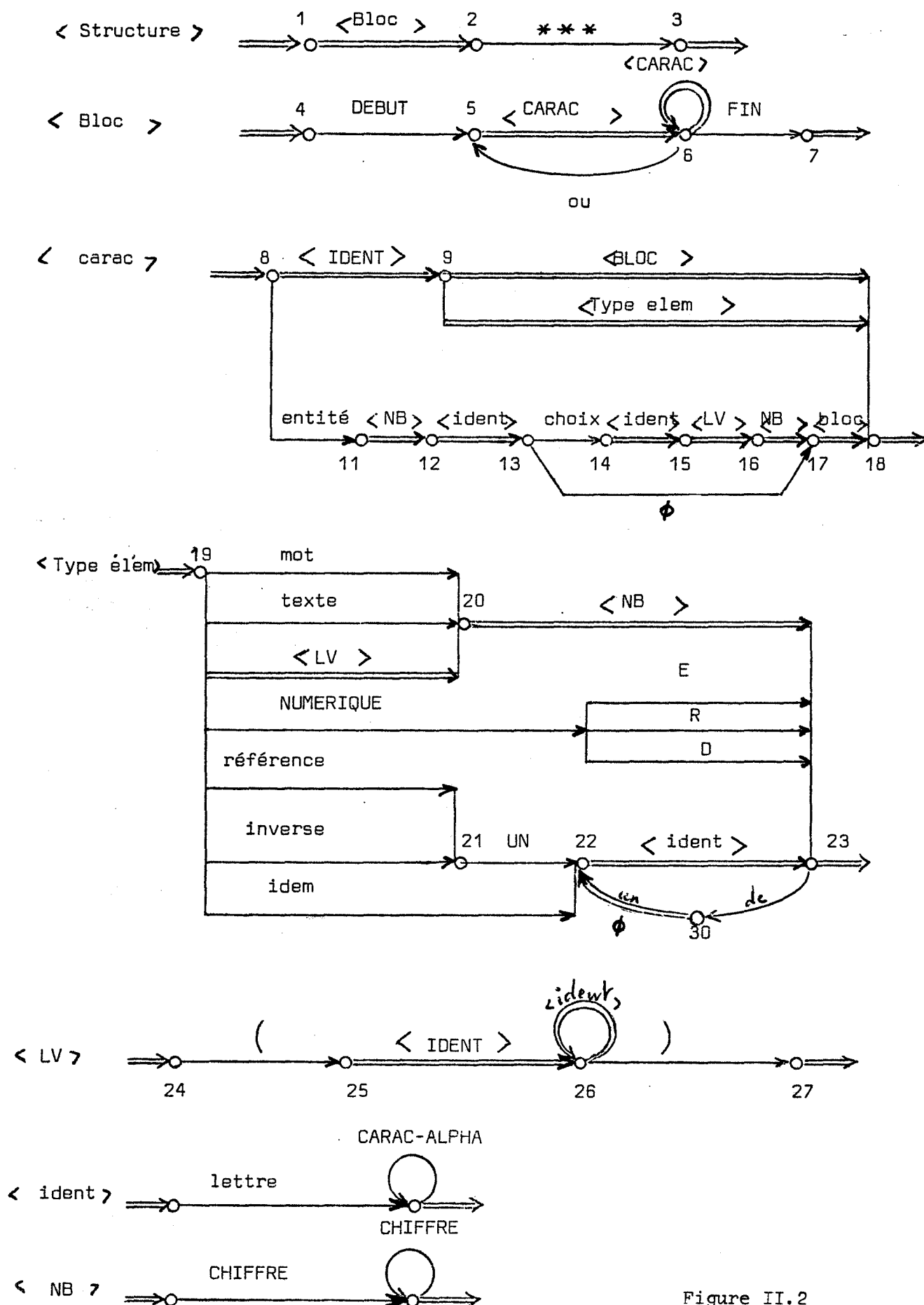


Figure II.2

La grammaire du langage de définition de structure peut être syntaxiquement définie en notation de Bacchus-Naur par les productions suivantes :

(rappel des notations dans la partie III-§ 3.1)

avec les différences suivantes:

() : symboles optionnels

[] : symboles obligatoires

< STRUCTURE > :: = < BLOC > ***

< BLOC > :: = DEBUT < CARAC > { < CARAC > | OU < CARAC > } FIN

< CARAC > :: = < IDENT > [< BLOC > | < TYPE-ELEM >] | ENTITE < NB > < IDENT >
(CHOIX < IDENT > < LV > < NB >) < BLOC >

< TYPE-ELEM > :: = [MOT | TEXTE | < LV >] < NB > | NUMERIQUE [E | R | D]
[[référence | INVERSE] UN | IDEM] < IDENT >
{ DE (UN) < IDENT > }

< LV > :: = (< IDENT > { < IDENT > })

< IDENT > :: = lettre { carac-alpha. }

< NB > :: = chiffre { chiffre }

lettre $\in \{ A, B, C, \dots, Z \}$

carac-alpha est un élément de l'ensemble des caractères alpha-numériques, chiffre $\in \{ 0, 1, 2, \dots, 9 \}$

Il existe des caractéristiques simples qui correspondent à un seul élément d'information (MOT, TEXTE.....) et des caractéristiques plus complexes qui groupent un certain nombre de caractéristiques. (BLOC, ENTITE)

Les blancs servent de séparateurs entre les mots du langage et, en conséquence, les identificateurs ne doivent pas contenir de blanc. Le premier caractère d'un identificateur doit être une lettre. Seuls les 16 premiers caractères sont pris en compte dans leur traitement.

2.2.2. - Description des caractéristiques :

Voici un exemple de structure arborescente, la description de cette structure en L.D.S. est donnée à la page suivante :

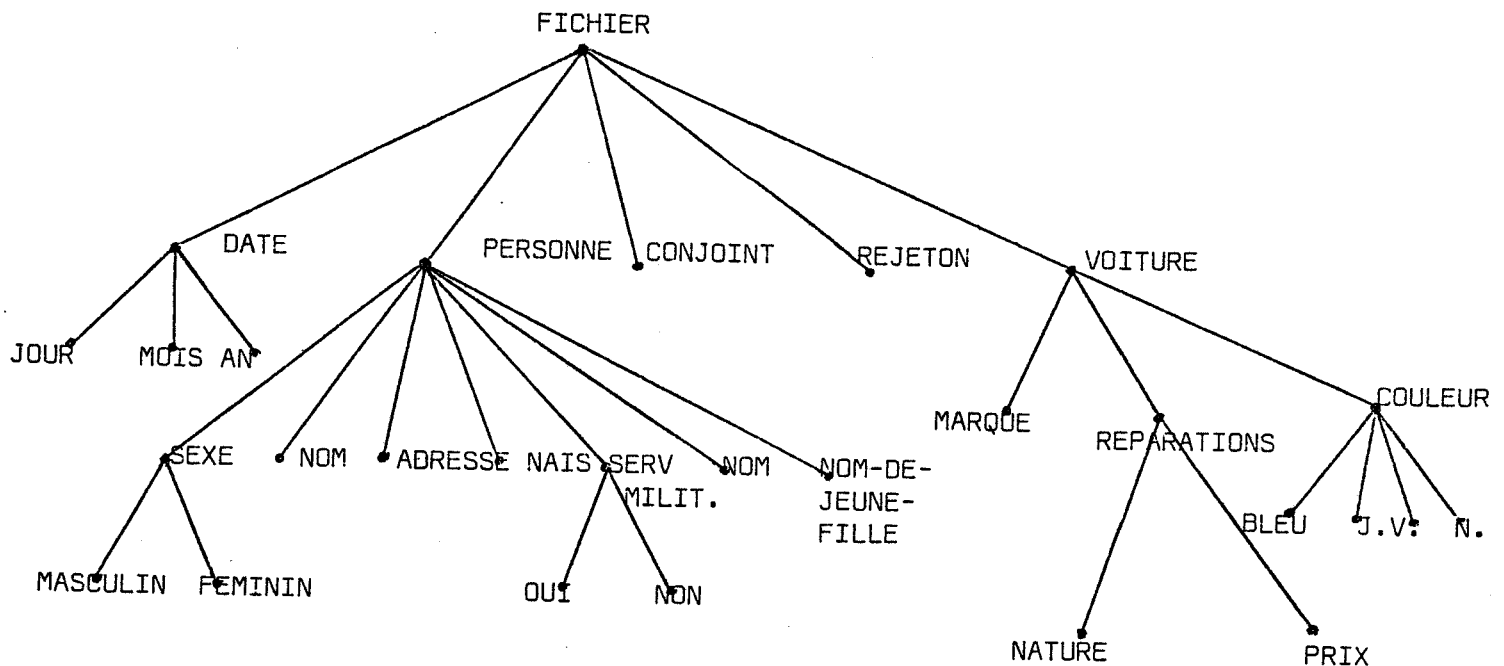


Figure II.3

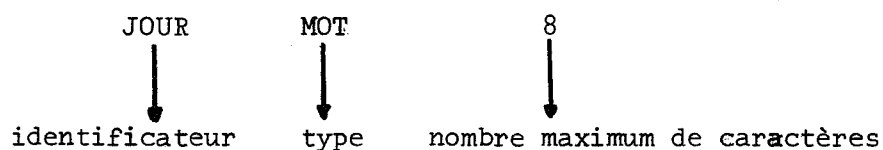
Dans la description ci-dessous les mots clés du langage ont été soulignés dans le but de les mettre en évidence .

EXEMPLE

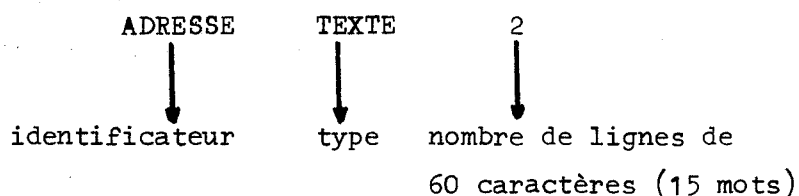
DEBUTDATE DEBUTJOUR MOT 8MOIS MOT 10AN NUMERIQUE EFINENTITE 10 PERSONNE CHOIX SEXE (MASCULIN FEMININ) 2DEBUTNOM MOT 10ADRESSE TEXTE 2NAISSANCE IDEM DATESERVICE-MILITAIRE (OUI NON) 2OUNOM MOT 10NOM-DE-JEUNE-FILLE IDEM NOMFINCONJOINT REFERENCE UNE PERSONNEREJETON INVERSE UNE PERSONNEENTITE 15 VOITUREDEBUTMARQUE MOT 12ENTITE 20 REPARATIONSDEBUTNATURE MOT 16PRIX NUMERIQUE RFINCOULEUR (BLEU JAUNE VERT NOIR) 11FINFIN ***

Les pages suivantes donnent quelques indications sur les différents types de caractéristiques.

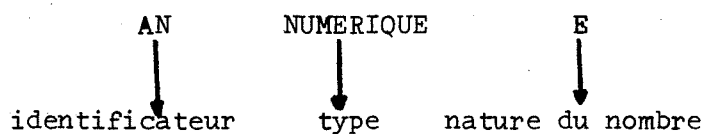
.../...

2.2.2.1. - TYPE MOT :

La valeur de jour sera une chaîne de caractères alphanumériques le premier étant une lettre. Le nombre maximum de caractères est le nombre suivant "MOT".

2.2.2.2. - TYPE TEXTE :

La valeur adresse sera une chaîne alphanumérique quelconque d'au plus 2 lignes soit 120 caractères.

2.2.2.3. - TYPE NUMERIQUE :

La valeur de AN sera un

- * nombre entier si la lettre suivant numérique est un E
- * nombre réel si la lettre suivant numérique est un R
- * nombre réel double-précision si la lettre suivant numérique est un D.

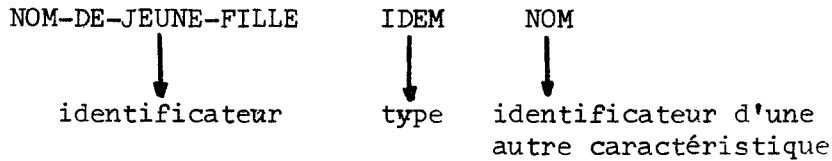
2.2.2.4. - TYPE LISTE DE VALEURS (LV)

COULEUR (BLEU JAUNE VERT NOIR) 11

.../...

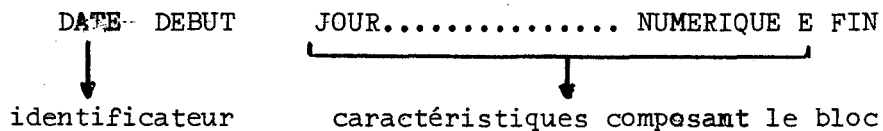
La valeur de couleur sera l'une des chaînes alphanumériques précisées dans la liste contenue dans le jeu de parenthèses. Ces chaînes sont séparées entre elles par des blancs. Dans cet exemple le nombre total des chaînes possibles dans une mise à jour ultérieure de la structure est 11

2.2.2.5. - TYPE IDEM :



la caractéristique NOM-DE-JEUNE-FILLE aura le même type que la 1ère caractéristique NOM de la structure c'est-à-dire sera un mot de 10 caractères.

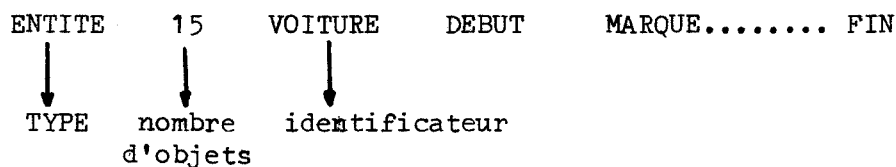
2.2.2.6. - TYPE BLOC :



DATE est un bloc constitué par les informations JOUR, MOIS, AN. Un BLOC est un ensemble de caractéristiques (pouvant être elles-mêmes du type BLOC) auquel est associé un identificateur.

L'examen des automates du LDS montre que toute structure est un BLOC. Il lui est associé un identificateur ici EXEMPLE .

2.2.2.7. - TYPE ENTITE :



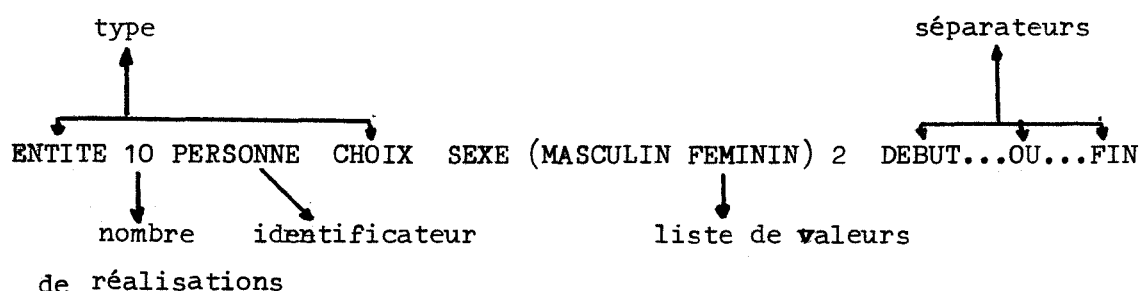
.../...

Une entité représente un ensemble de blocs ayant même la structure.

L'ensemble des informations associées à un de ces blocs est appelé réalisation de l'entité.

L'entité VOITURE permet de stocker 15 réalisations numérotées de 1 à 15 et contenant chacune les informations MARQUE, REPARATIONS, COULEUR. On considère, par exemple, l'information : MARQUE DE LA VOITURE 7

2.2.2.8. - TYPE ENTITE CHOIX :

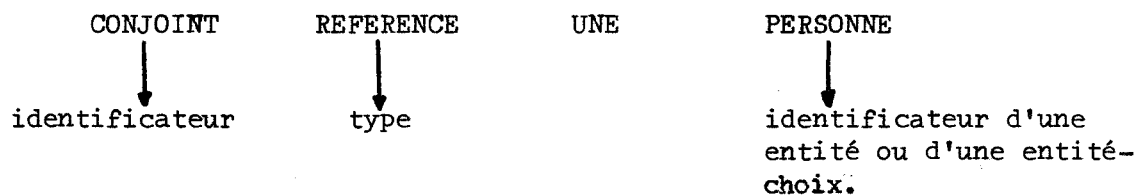


Une entité choix représente un ensemble de blocs.

Chaque réalisation de l'entité correspond à un bloc qui comprend la caractéristique liste de valeurs placée après le mot CHOIX et, suivant la valeur de cette caractéristique, les informations indiquées soit entre DEBUT et OU, soit entre deux OU successifs, soit entre OU et FIN.

Dans l'exemple, les réalisations pour lesquelles la caractéristique SEXE aura la valeur MASCULIN comprendront les informations NOM, ADRESSE, NAISSANCE, SERVICE-MILITAIRE, tandis que celles pour lesquelles SEXE aura la valeur FEMININ comprendront les informations NOM et NOM-DE-JEUNE-FILLE.

2.2.2.9. - TYPE REFERENCE :



.../...

CONJOINT désigne une réalisation de l'entité PERSONNE.

Une caractéristique de ce type ne peut faire référence qu'à une caractéristique définie antérieurement dans la structure. Elle n'a de valeur que si la caractéristique citée en a déjà une.

2.2.2.10. - TYPE INVERSE :



La valeur de REJETON sera une chaîne de bits de dimension égale au nombre de réalisations possibles de l'entité PERSONNE. La *i*ème réalisation de l'entité PERSONNE possèdera la propriété REJETON si la *i*ème bit de la chaîne inverse est à 1.

3 - ANALYSE SYNTAXIQUE ET COMPILATION DE LA STRUCTURE CREEE:

La structure décrite à l'aide du LDS sert de données à une procédure appelée STRUCTUR.

Le résultat de ce programme (outre la détection des erreurs syntaxiques) est la construction du fichier structure. Ce fichier est formé d'une suite de vecteurs, chacun correspondant à un identificateur de la structure écrite en L.D.S.

L'ensemble des fonctions et des pointeurs contenus dans chaque vecteur permet de déterminer l'adresse absolue dans le fichier données de la caractéristique cherchée. Il n'y a pas de recherche dans ce dernier fichier qui est très grand, celle-ci se fait dans le fichier structure, qui par sa dimension, reste presque continuellement en mémoire centrale lors d'une session.

Chaque vecteur a la composition suivante :

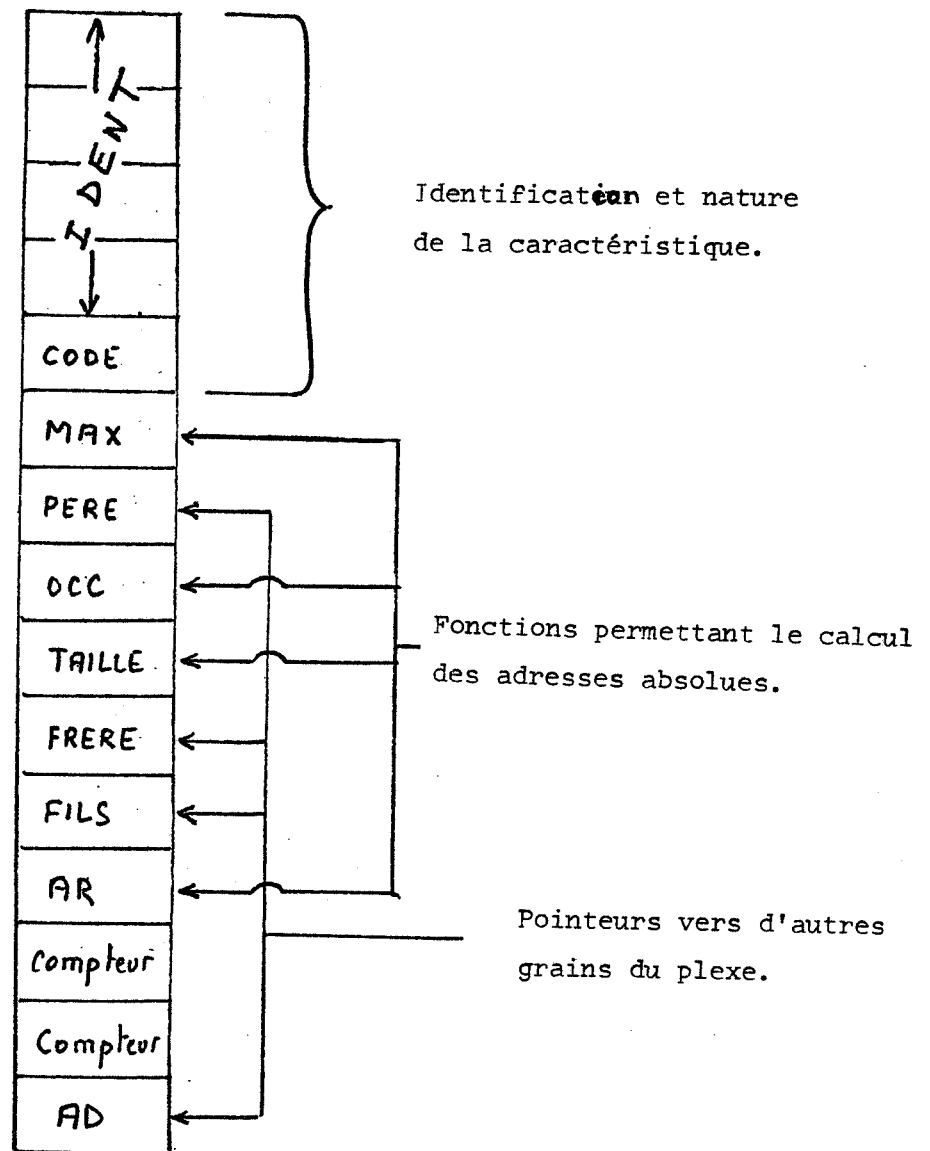


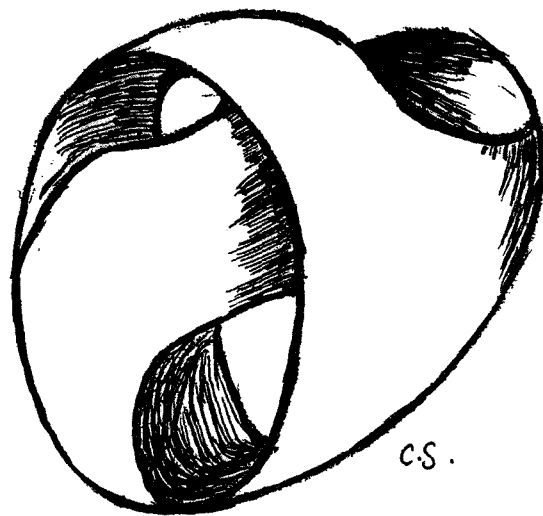
Figure II.4

4 - LES ACCES AUX BASES SOMINE :

Cette quatrième phase du travail de création et d'utilisation d'une base SOMINE constitue la relation entre les utilisateurs et la base. Elle fait l'objet de la partie III de ce rapport.

PARTIE III

LES ACCES AUX BASES SOMINE



SOMMAIRE

	Page
1 - INTRODUCTION :	
Qualités souhaitables pour la création et l'utilisation de SOMINE	III- 3
1- 1 Accès à la banque	3
1- 2 Procédure d'accès à la banque	4
2 - MANIPULATION DES FICHIERS :	4
2- 1 Quelques remarques sur l'arbre-structure	5
2- 2 Définition des noeuds de l'arbre-structure	6
2- 3 Cheminement dans la structure arborescente	7
2- 4 Utilisation du fichier-données	10
3 - MOYEN D'ACCES A LA BANQUE : Spécification de la requête	11
3- 1 Syntaxe de la requête	11
3- 2 Sémantique de la requête	13
3.2.1. - Le mode	13
3.2.2. - Les démonstratifs	15
3.2.3. - La citation	15
3.2.4. - La valeur	17
3.2.5. - Exemples de requêtes	17
4 - COMMENT LA REQUETE REpond AUX QUALITES SOUHAITEES POUR LE TRAITEMENT DES INFORMATIONS .	18
5 - MISE EN OEUVRE DE L'ACCES A LA BASE	19
5- 1 Sous-programme d'analyse et de pré-interprétation	21
5.1.1. - Les fonctions de ce sous-programme	21
5.1.2. - Exemple	22

5- 2	Sous-programme d'interprétation	III. 23
5.2.1.	- Détermination de l'adresse structurelle	25
5.2.2.	- Calcul de l'adresse absolue (niveau non terminal)	26
5.2.3.	- Calcul de l'adresse absolue (niveau terminal)	27
6 -	APPLICATION :	30
6- 1	Exemple de programme utilisateur	30
6- 2	Les résultats obtenus	31
6- 3	Remarques	31
6- 4	Exemples de programmes standard	32
6- 5	Remarques relatives à la programmation	33
7 -	AMELIORATIONS APORTEES A L'INTERPRETEUR DE BASE :	35
7- 1	Requêtes groupées	36
7- 2	Requêtes booléennes	37
7- 3	Synonymie	39
7- 4	Les requêtes contenant le séparateur "TOUT"	40
7- 5	Requêtes à valeur variable	42
7- 6	Requêtes discriminantes	42
7- 7	Schéma général	43
8 -	MULTI-ACCES A LA BASE	44
9 -	PROBLEMES LIES A LA GESTION DES UTILISATEURS	44
10 -	CONCLUSION	44
11 -	BIBLIOGRAPHIE	47

Lorsque l'implémentation des structures a été possible, nous avons pensé au problème de l'utilisation de SOMINE .Pour le résoudre une étude documentaire et comparative de différentes bases de données a été entreprise/1/,/2/,/3/,/4/,/5/,/6/,/19/ . Ceci nous a permis de connaître quelques réalisations existantes et de formuler les options souhaitables pour le futur système de bases de données SOMINE . Les critères que nous avons retenus ont souvent été dictés par les types d'applications susceptibles d'utiliser une base de données . Nous avons mis en oeuvre une application dans chacun des domaines trouvés .Le matériel sur lequel nous avons réalisé SOMINE (P1175 de l'Ecole Nationale Supérieure des Mines de Saint-Etienne) permet le traitement des programmes en multiprogrammation , les terminaux disponibles au Centre sont : une imprimante, un lecteur de cartes, des unités de disques et de bandes magnétiques ainsi que des écrans alphanumériques . C'est dans cet environnement que nous avons réalisé les accès aux bases de type SOMINE ./18/.

1. - QUALITES SOUHAITABLES POUR LA CREATION ET L'UTILISATION DE SOMINE :

Nous avons voulu que l'accès à la banque soit souple et adapté à l'utilisation que l'on veut en faire, pour cela nous nous sommes attachés à varier les accès à la base et à simplifier la procédure permettant ces accès.

1.1. - Accès à la base

Nous avons permis trois types d'accès :

a) Accès conversationnel : Ceci pour permettre une mise en oeuvre facile et une sortie immédiate des résultats. Cette méthode rend possible un travail au coup par coup c'est-à-dire dans lequel le résultat d'un accès peut orienter le ou les accès suivants. Cette méthode laisse aussi la possibilité de corriger les erreurs de l'utilisateur au fur et à mesure du déroulement des requêtes. L'écran permet aussi le traitement des informations en temps réel et à distance.

.../...

b) traitement par lots : Il est valable pour toute utilisation mais il convient plus particulièrement à un travail important ou pour un accès effectué dans le but de recueillir une ou plusieurs informations qui subiront des traitements extérieurs à la base (ces traitements pourront d'ailleurs être entrepris aussitôt que l'accès est réalisé, voir § 4).

c) Accès intégrés au programme de l'utilisateur, ceci permet de constituer une bibliothèque avec les programmes dont l'usage est fréquent. Par exemple, des programmes de mise à jour ou d'interrogation standards dont l'exécution peut être lancée à tout moment.

1.2. - Procédure d'accès à la base :

Afin de simplifier nous avons voulu qu'il soit possible d'accéder à la banque à partir de n'importe quel programme écrit en langage évolué. Ainsi un utilisateur ne connaissant que ces langages pourra facilement apprendre à se servir de SOMINE. Nous espérons ainsi encourager les utilisateurs et les rendre opérationnels en un temps minimum.

2. - MANIPULATION DES FICHIERS :

Après avoir parlé des qualités qui nous paraissent nécessaires pour une utilisation souple et simple de SOMINE, le problème qui se pose est la manipulation des fichiers en vue d'obtenir la meilleure utilisation possible de la structure mise en place.

Les méthodes utilisant des index comme en documentation automatique /6/ ou le chaînage des enregistrements des fichiers n'utilisent pas les possibilités de la structure définie. En effet, la structure peut être représentée par un graphe orienté, c'est-à-dire que nous avons privilégié un certain nombre de relations entre les éléments qui la constituent. En particulier, la relation binaire "fils de " ou la relation inverse "père de" constituent les articulations de la structure SOMINE. Afin d'utiliser les possibilités de la structure analysons les différentes façons de se déplacer dans l'arbre structure et la meilleure méthode pour accéder à un noeud donné.

L'ensemble des renseignements qui ont été retenus dans chaque grain du fichier-structure permet par accès à ce fichier, de déterminer l'adresse virtuelle de l'information considérée, et à travers la gestion de

mémoire, il y a possibilité de déterminer l'adresse du fichier-données contenant une information ou destinée à en recevoir une. L'important est donc de retrouver dans le fichier-structure le noeud de l'arbre qui désigne l'information en question.

2.1. - Quelques remarques sur l'arbre structure :

L'arbre tel qu'il a été décrit (part. II) représente la structure mais il ne représente pas exactement les informations du fichier-données. En effet, les champs composés peuvent avoir plusieurs réalisations (entité ou entité-choix) les noeuds de l'arbre qui les représentent devront être dupliqués autant de fois qu'il peut y avoir de réalisations.

Deux cas sont à distinguer :

- dans le cas d'une entité, les sous-arbres fils de chaque réalisation sont structurellement identiques (figure 1).

- dans le cas d'une entité-choix, plusieurs structures sont possibles pour le sous-arbre fils de chaque réalisation. Le choix est déterminé par la valeur de la caractéristique discriminante de cette entité qui est la liste de valeurs choix (figures 2 et 3)

Les exemples suivants font référence à la structure de la partie II

Cas de l'entité réparation qui a 20 réalisations possibles.



Figure III-1

Cas de l'entité choix personne

Il y a dix sous arbres de racine PERSONNE et ils ont l'une des deux structures suivantes :

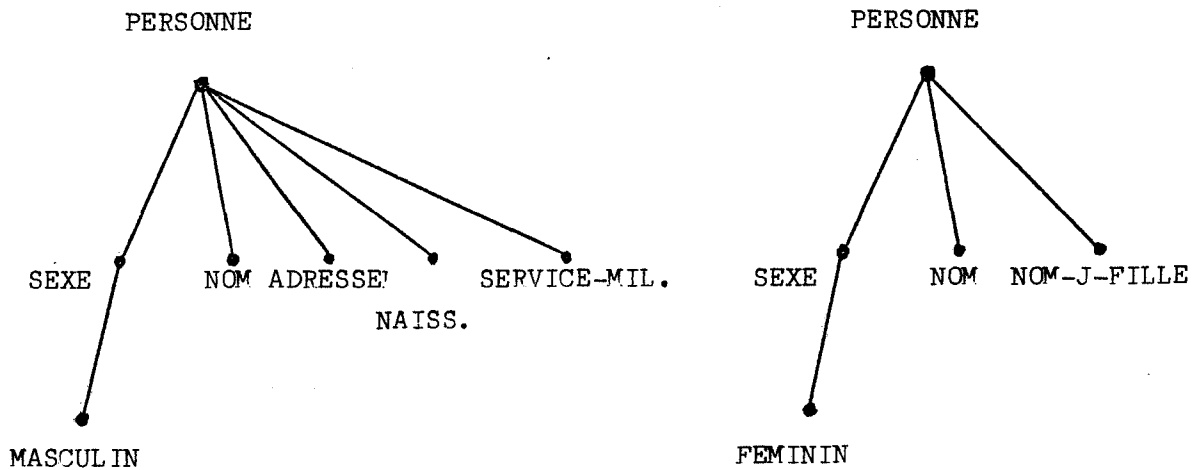
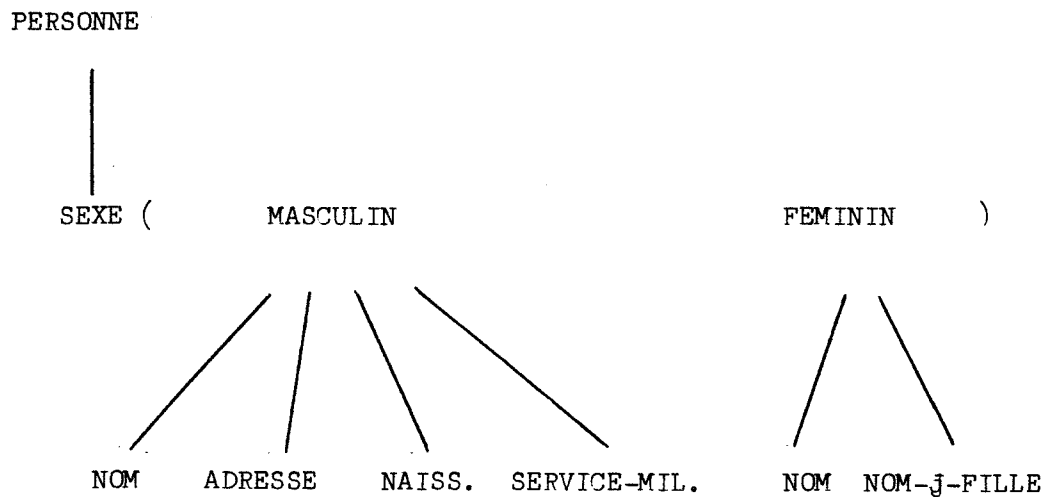


Figure III-2

L'entité choix PERSONNE peut également être représentée par le schéma suivant :

Figure III-3



2. 2 - Définition des noeuds de l'arbre structure :

Ces exemples montrent que chaque noeud de l'arbre peut ne pas être défini de façon unique

- Dans le cas des caractéristiques simples et des blocs, la désignation définit de façon unique le noeud de l'arbre correspondant.

- Dans le cas des entités, le couple (désignation, numéro de réalisation) définit complètement un noeud de l'arbre. Si dans la désignation d'une entité le numéro de réalisation est absent, SOMINE peut décider de sa valeur, l'option retenue est la consultation de la chaîne de bits d'existence de l'entité, le numéro retenu est celui du premier bit qui répond à la question.

- Dans le cas d'une entité-choix, c'est la connaissance du triplet (désignation, numéro de réalisation, occurrence) qui définit parfaitement une réalisation. Lors de la création, il y a attribution d'un numéro de réalisation et ensuite, mise à jour de l'occurrence. Ceci montre qu'après la création, la connaissance du numéro de réalisation détermine la valeur de l'occurrence, les deux derniers éléments du triplet ne sont pas indépendants. L'occurrence peut-être considérée comme un filtre au niveau de la structure. Si la définition de la caractéristique discriminante (LV-choix) sur laquelle repose le choix structurel de l'entité-choix est faite, ensuite le couple (désignation, numéro de réalisation) caractérise aussi de façon unique une réalisation d'entité-choix.

2. 3 - Cheminement dans la structure arborescente :

Les remarques précédentes montrent que la méthode de cheminement recherchée pour accéder à un noeud de l'arbre doit citer tous les couples (désignation, numéro de réalisation) des noeuds rencontrés sur le chemin d'accès au noeud considéré. Ceci afin de caractériser sans ambiguïté les noeuds de la structure et leur chemin d'accès, ce qui évitera les retours arrière au cours de la recherche et la dégradation des performances qui s'en suivrait .

La méthode de cheminement retenue est :

La promenade dans la structure par la méthode "depth-first" /11/
=====

Elle consiste à accéder le plus vite possible à la caractéristique dont il est question. Pour cela, il y a descente dans la structure depuis la racine jusqu'au noeud considéré, un seul chemin relie un noeud quelconque de l'arbre à la racine. Pour effectuer le cheminement voulu avec rapidité et sans retour en arrière dans la recherche, il faudra désigner tous

les noeuds intermédiaires du chemin à parcourir et dans l'ordre dans lequel ils doivent être rencontrés. Il s'agit d'une citation hiérarchique : la caractéristique cherchée devra être citée suivie de son père du père de celui-ci etc.. jusqu'à la racine de l'arbre ancêtre de toutes les caractéristiques. Un noeud de l'arbre représente une caractéristique mais celle-ci peut avoir plusieurs réalisations (par exemple les entités) afin d'indiquer sans ambiguïté, quelle réalisation est considérée, afin aussi de déterminer si une caractéristique citée est un filtre (caractéristique inverse) des séparateurs entre les différents identificateurs semblent nécessaires pour définir avec précision la citation hiérarchique.

La promenade dans la structure par cette méthode utilise en particulier les fonctions d'accès au frère et au fils de chaque caractéristique. Elle peut se représenter par l'organigramme général suivant, dans lequel I désigne l'identificateur d'un noeud, les variables FRERE (I), PERE (I), et FILS (I) désignent le frère, le père et le fils de l'identificateur I .

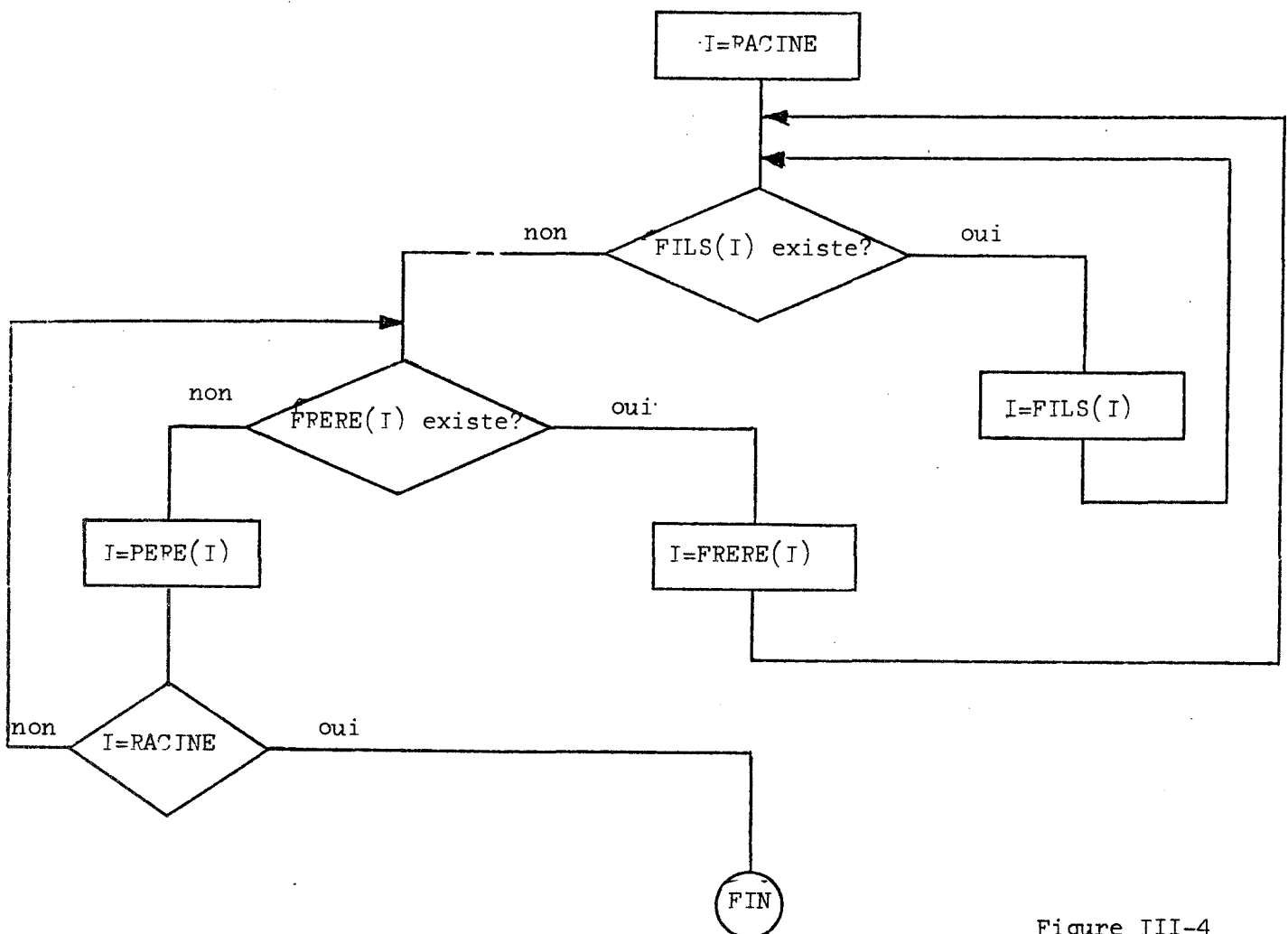


Figure III-4

remarque :

L'organigramme précédent permet de parcourir tous les noeuds de l'arbre, si l'on veut accéder à un noeud particulier, il y a utilisation de cet organigramme mais à chaque niveau de l'arbre, il y a sélection du sous-arbre dans lequel va se poursuivre la recherche.

exemple : En se reportant toujours à l'exemple test supposons que nous cherchions :

LE PRIX D'UNE REPARATION D'UNE VOITURE

Le chemin parcouru par la méthode "depth-first" est celui indiqué par le schéma suivant qui représente une partie de la structure exemple :

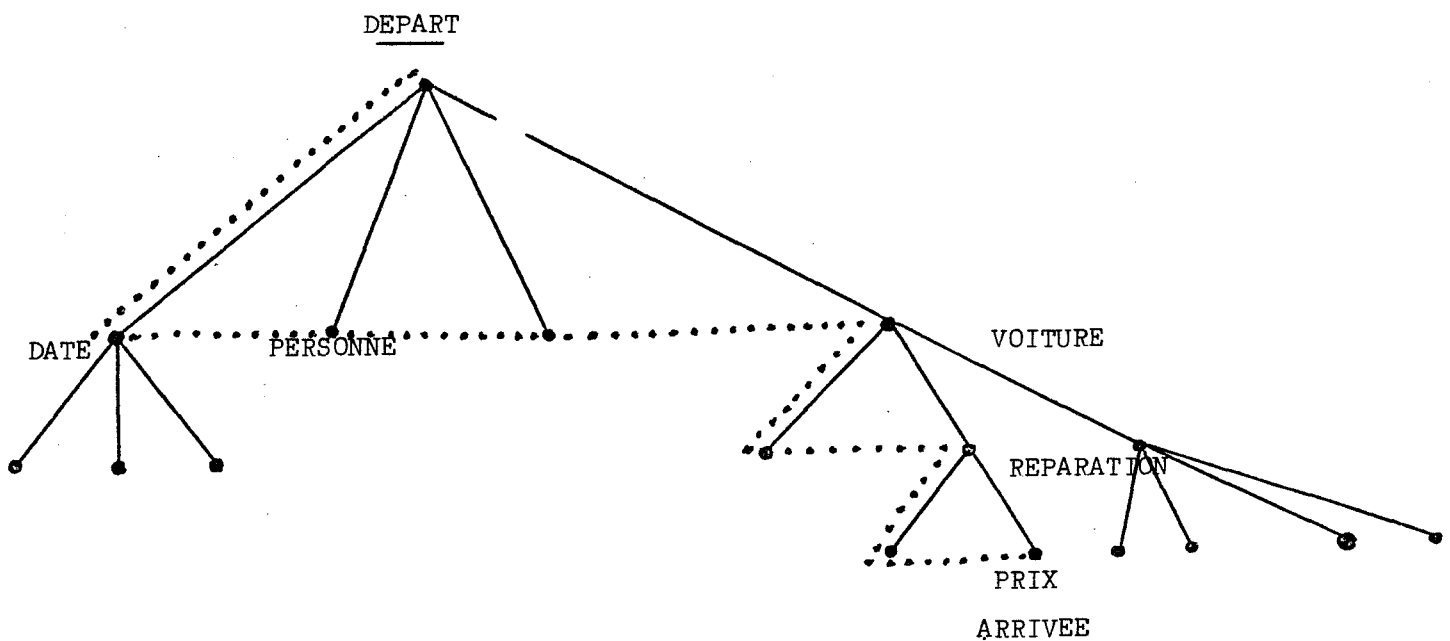


Figure III-5

Nous remarquons que cette méthode ne peut pas être mise en oeuvre par un utilisateur qui ne connaît pas bien la structure, en effet une erreur dans la citation hiérarchique fera du chemin une impasse.

Pour faciliter l'utilisation de SOMINE, il nous a semblé souhaitable de permettre l'accès à la structure. Cet accès peut s'envisager de plusieurs manières suivant les renseignements que désire l'utilisateur.

Ce qui a été dit sur le cheminement "depth-first" nous montre que la connaissance des identificateurs qui désignent les caractéristiques n'est pas suffisante, encore faut-il connaître le type de ces identificateurs ainsi que les bornes supérieures attribuées aux caractéristiques simples. En somme, la structure telle qu'elle a été décrite en L.D.S. n'est pas redondante de plus elle représente la structure dans un langage connu des utilisateurs de SOMINE. Nous avons donc laissé la possibilité de connaître la structure en L.D.S. avant une séance de travail car elle constitue une aide à l'accès à la base, c'est la table des matières de la banque. Cette structure pourrait également renseigner l'utilisateur sur son droit ou non d'accéder à telle partie de la structure ce qui permet une première évocation des problèmes de protection de l'information dans les bases de données.

2. 4 - Utilisation du fichier-données :

Il n'a pas été utilisé lors de l'implémentation de la structure, c'est pendant l'utilisation de SOMINE que l'on a accès à ce fichier. Il contient les informations de la base à des adresses qui ont été définies (et peuvent être retrouvées) pendant la mise en place de la structure choisie pour la base.

La méthode de cheminement qui a été décrite permet de retrouver une caractéristique de la base, elle est alors repérée par son adresse structurelle, c'est-à-dire l'adresse de la sous-page de mémoire virtuelle qui contient tous les renseignements relatifs à cette caractéristique, (fichier structure). Pour chaque étape du chemin décrit qui joint la racine de l'arbre à la caractéristique recherchée, nous connaissons les fonctions d'accès et différents pointeurs /16/

Voyons comment la connaissance de ces informations nous permet de retrouver (ou de placer) une information dans le fichier-données. La connaissance de l'adresse relative (par rapport au père d'une caractéristique) permet de calculer l'adresse absolue du début d'une caractéristique si celle du père est connue. Lorsqu'il s'agit d'une entité, la taille d'une réalisation et l'adresse absolue du début de la réalisation permettent d'accéder à une réalisation donnée.

La connaissance de l'adresse absolue 0 qui est celle du début de la structure amorce un processus itératif qui détermine l'adresse absolue de chaque étape du cheminement, et permet enfin d'obtenir l'adresse de l'étape

d'arrivée (adresse par rapport au début du fichier-données). Chaque mise en relation avec la base ne demande outre les recherches de bits d'existence qu'un seul accès au fichier-données, celui-ci se faisant à l'adresse absolue déterminée par l'algorithme précédent. Le fichier-données peut - être de très grande dimension, il n'y a pas de recherche dans ce fichier, l'approche du noeud en question se fait dans le fichier-structure qui est de petite dimension. On peut donc espérer un gain de temps appréciable dans la recherche. De plus, les différents accès à la structure ne demanderont pas beaucoup d'accès au disque la contenant, en effet, le plan primaire de la gestion de mémoire (enregistrement physique) se compose de quatre pages en mémoire centrale, ce qui est en général supérieur à la taille d'une structure moyenne.

3. - MOYEN D'ACCES A LA BANQUE - SPECIFICATIONS DE LA REQUETE :

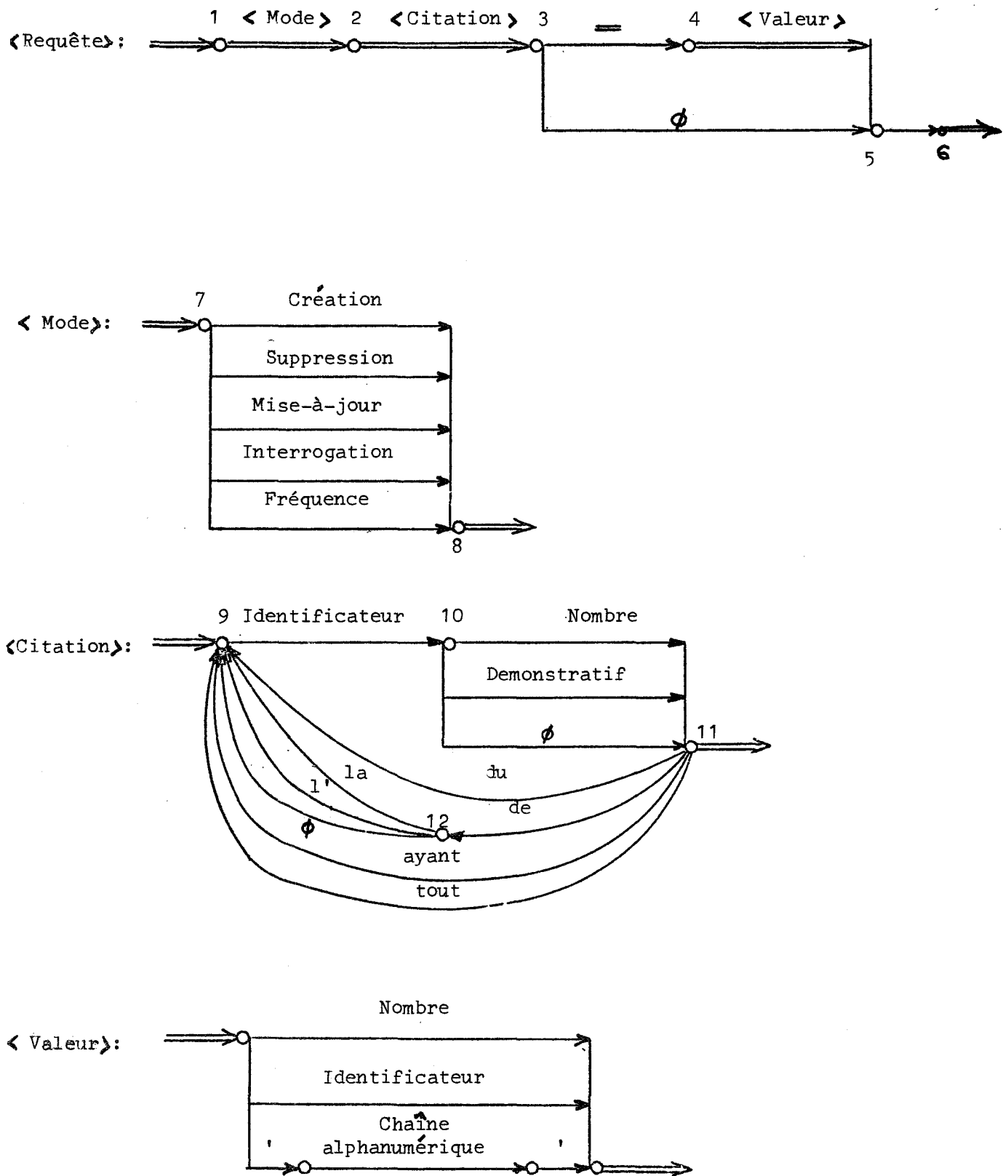
Nous avons vu que lors d'un accès à la banque, si l'on veut éviter les retours en arrière, il est nécessaire de nommer les étapes successives du chemin de recherche. C'est la requête qui réalise ce balisage, toute mise en relation avec la base se fera par ce moyen souple et facile à mettre en oeuvre. Son rôle est de détecter l'adresse où se trouve l'information, de vérifier sa validité ainsi que celle de l'action à entreprendre et enfin de réaliser cette action. Durant une session l'utilisateur pourra se servir d'autant de requêtes qu'il le désire.

3.1. - Syntaxe de la requête :

Une requête est définie par les automates de la page suivante dans lesquels les blancs servent de séparateurs entre les différents éléments.

remarque : La requête peut être définie par un seul automate, dans le schéma de la page suivante, il a été volontairement décomposé en plusieurs sous-automates pour en faciliter l'explication sémantique.

Figure III-6 : LES AUTOMATES DE LA REQUETE



Si l'on adopte la métalangue de Bacchus-Philips (dérivée de la forme normale de Bacchus Naur) pour représenter la requête on obtient la description suivante :

$\langle \text{requête} \rangle :: = \langle \text{mode} \rangle \langle \text{citation} \rangle \left[\underline{=} \langle \text{valeur} \rangle \right] \#$
 $\langle \text{mode} \rangle :: = \underline{\text{création}} \mid \underline{\text{suppression}} \mid \underline{\text{mise à jour}} \mid \underline{\text{interrogation}} \mid \underline{\text{fréquence}}$
 $\langle \text{citation} \rangle :: = \langle \text{identificateur} \rangle \langle \text{nombre} \rangle \mid \underline{\text{X}}(\underline{n}) \left[\left[\underline{\text{du}} \mid \underline{\text{de}} \mid \underline{\text{la}} \mid \underline{\text{de l'}}
 $\langle \text{valeur} \rangle :: = \langle \text{nombre} \rangle \langle \text{identificateur} \rangle \mid \langle \text{chaîne alphanumérique} \rangle \mid \langle \text{nombre} \rangle . \langle \text{nombre} \rangle$
 $\langle \text{n} \rangle :: = \langle \text{nombre} \rangle \langle \text{identificateur} \rangle$$

Les symboles de langage sont soulignés

Les productions $\langle \text{identificateur} \rangle$ et $\langle \text{nombre} \rangle$ ont déjà été définies en II.7

Rappels : $\left[\right]$ symboles optionnels
 $/11/$
 $\left[\right]$ symboles obligatoires
 $\{ \}$ symboles d'omission ou de répétition un nombre quelconque de fois
 $|$ symboles "OU"
 $::=$ symbole d'affectation

3.2 - Sémantique de la requête :

3.2.1. - Le mode

Il caractérise la nature du traitement que l'on va faire subir à l'information qui va être citée. A l'intérieur d'une banque de données les informations doivent pouvoir naître, vivre et mourir, les quatre modes de requêtes de base le permettent.

* création et suppression :

Ce sont deux modes inverses. Ils ne s'appliquent qu'aux entités et aux entités-choix, ainsi qu'aux caractéristiques de type référence. On peut également créer ou supprimer une caractéristique de type idem si elle cite une caractéristique des types précédents. Le but de ces modes est de faire naître ou mourir les caractéristiques c'est - à - dire de réserver ou de libérer la place physique

qui leur est nécessaire. On peut déclarer un grand nombre de réalisations d'une même entité, la place n'est réellement réservée sur le fichier-données que lors de la création des réalisations. Concrètement, pour les entités et les entités choix, il y a mise à 1 du bit d'existence correspondant à la réalisation demandée, ceci dans le cas de la création et mise à 0 de ce même bit dans le cas de la suppression. Si la caractéristique est de type référence, il y a addition ou soustraction d'un maillon à la chaîne des références. Si une caractéristique doit être créée, la création intervient avant toute action sur ses descendants.

La création d'une réalisation d'entité peut affecter un numéro à cette réalisation si elle n'en avait pas, ce sera son numéro d'ordre et c'est aussi celui du premier bit nul de la chaîne d'existence.

* Mise à jour et interrogation :

Ces deux modes assurent la vie d'une information en lui permettant d'entrer et de sortir de la base. La mise à jour fait entrer une information dans la base à la place qui lui est réservée, l'interrogation permet de sortir une information déjà contenue dans la base. Ces deux modes ne sont utilisés que pour les caractéristiques terminales : mot, texte, liste de valeurs, valeur numérique, inverse.

Remarque :

La mise à jour de la liste de valeurs choix d'une entité-choix doit être faite avant toute autre mise à jour car elle détermine l'occurrence de la réalisation /16/

* Fréquence :

C'est un mode de contrôle des informations, il permet de connaître la valeur d'un pointeur associé à chaque noeud de l'arbre-structure, la valeur de ce pointeur indique le nombre d'accès à cette caractéristique. Ce nombre permet de comparer les noeuds de même niveau et peut-être de modifier la structure afin d'améliorer le temps d'accès à une information dont la fréquence est élevée. Ce mode peut également servir à faire des statistiques sur l'utilisation des différentes caractéristiques de la base et permettre une optimisation de la structure des informations.

* Autres requêtes :

Il suffit d'ajouter une transition entre les états 7 et 8 de l'automate mode pour créer de nouvelles requêtes si le besoin s'en fait sentir, par exemple pour une application qui ferait subir un traitement particulier aux informations contenues dans la base.

Remarque :

Il n'est pas nécessaire que les mots CREATION, SUPPRESSION, MISE A JOUR, INTERROGATION, FREQUENCE ... figurent en entier dans une requête, la reconnaissance du mode se faisant sur la première lettre C, S, M, I, F...

3.2.2. - Les démonstratifs :

Ils interviennent dans l'automate citation (page 13), ce sont des variables entières indicées notées X (I) dans lesquelles I est une variable entière ou une constante entière. Un démonstratif sert à désigner le numéro de réalisation de la caractéristique qui le précède. Il ne se trouve donc, qu'après la désignation d'une entité ou d'une entité-choix.

L'utilisation des démonstratifs présente deux avantages :

- elle permet d'utiliser un numéro de réalisation sans en connaître la valeur, par exemple, lorsque ce numéro a été affecté à la variable X (I) lors de la création de cette réalisation où lorsque ce numéro est calculé par le programme d'utilisation.

- elle permet de faire exécuter plusieurs fois la même requête sans avoir à la reformuler. Il suffit de changer entre chaque exécution la valeur affectée au démonstratif pour faire subir le même traitement à plusieurs réalisations d'une même caractéristique (analogie avec les boucles DO en FORTRAN)

3.2.3. - La citation :

C'est la partie de la requête qui permet de citer la caractéristique que l'on recherche et toutes les étapes du cheminement qui doit être fait dans la structure.

Les identificateurs se succèdent dans la citation, ils sont isolés par des séparateurs et éventuellement des démonstratifs. L'automate de

la citation a une transition de l'état de sortie vers l'état d'entrée qui correspond au passage d'un niveau hiérarchique au niveau supérieur dans la structure. Pour permettre la transition entre les états 11 et 9, plusieurs séparateurs sont possibles :

- séparateur "de" : il est utilisé avant un identificateur de bloc, c'est-à-dire lorsque la filiation est unique.

- séparateurs "de la", "du", "de l'" : ils précèdent un identificateur (féminin ou masculin) d'entité ou d'entité-choix, celui-ci devra alors être suivi de son numéro de réalisation afin de désigner le sous-arbre de la structure dans lequel la recherche va être poursuivie.

- séparateur "tout", "toute" : il précède également une désignation d'entité ou d'entité-choix (non suivie d'un numéro de réalisation) car il permet de désigner toute les réalisations d'une même entité.

- séparateur "ayant" : c'est le filtre qui précède une caractéristique de type inverse.

Au niveau le plus bas de la citation, l'identificateur n'est suivi d'un démonstratif ou d'un nombre que s'il désigne une entité, sinon il y a passage à l'état 11 de l'automate de citation.

remarque : Dans le cas d'une entité et pour une requête création (ou suppression) les trois formes suivantes sont possibles :

C VOITURE 2 DE LA PERSONNE 2 #

C VOITURE DE LA PERSONNE 2 #

C VOITURE X(3) DE LA PERSONNE 2 #

La deuxième de ces requêtes crée la première réalisation non encore créée de l'entité personne (il y a recherche du premier bit égal à 0 dans la chaîne de bits d'existence).

La troisième de ces requêtes crée la réalisation dont le numéro est la valeur de X (3), ceci dans le cas où X (3) a été défini, dans le cas contraire (X (3) = 0) on affecte à X (3) une valeur égale au numéro d'ordre de la place de la première réalisation non encore créée.

3.2.4. - La valeur :

Dans une requête la valeur est précédée du signe "=" et suivie du dièse qui termine toute requête. La valeur n'apparaît que dans les requêtes de mise à jour des caractéristiques simples, elle sert alors de véhicule à l'information qui doit entrer dans la base. Afin de pouvoir envisager un éventail assez large d'applications pour les bases de type SOMINE, nous avons tâché de diversifier au maximum le type des valeurs possibles.

Dans l'interpréteur de base, les valeurs peuvent être :

- des identificateurs : c'est-à-dire des chaînes alphanumériques ne contenant pas de blancs et dont le premier caractère est une lettre. La taille maximum d'un identificateur est de 15 mots mémoire, soit 60 caractères.
- des listes de valeurs : la valeur est alors un identificateur choisi parmi ceux définis par la structure.
- des textes : chaînes alphanumériques délimitées par deux apostrophes elles peuvent contenir tout symbole même le blanc et l'apostrophe (celle-ci est alors doublée). Cette valeur permet de stocker dans la banque non seulement des "textes", mais également des dessins ou des graphes.
- des valeurs numériques : elles peuvent être positives ou négatives, entières, réelles simple précision ou double précision. Ce sont les limites du calculateur sur lequel est implantée la base qui limitent les valeurs que l'on peut leur attribuer. Elles peuvent être précédées ou non d'un signe et c'est le point décimal qui sépare la partie entière de la partie décimale dans le cas des valeurs réelles ayant une partie décimale.
- on peut envisager aussi de faire figurer comme valeur une chaîne alphanumérique qui représente une instruction ou un programme écrit en Fortran ou en Cobol. On verra dans la partie IV une solution à ce problème.

3.2.5. - Exemples de requêtes :

```

M JOUR DE DATE = LUNDI #
C PERSONNE 2 #
M SEXE DE LA PERSONNE 3 = FEMININ #
I ADRESSE DE LA PERSONNE X(4) #
I PRIX DE LA REPARATION 1 DE LA VOITURE 2 #

```

```

M    COULEUR DE LA VOITURE X(5) = JAUNE  #
S    VOITURE 10      #
F    MOIS DE DATE   #

```

4 - COMMENT LA REQUETE REpond AUX QUALITES SOUHAITEES POUR LE TRAITEMENT DES INFORMATIONS :

La requête est un moyen simple d'accéder à la base, elle se présente comme une chaîne alphanumérique qui devra être analysée et interprétée. Une telle chaîne peut être considérée comme une constante "hollerith" et devenir l'argument du sous-programme Fortran qui va faire cette analyse et cette interprétation. Cette constante hollerith peut aussi être considérée comme une donnée d'un programme utilisateur, elle est alors introduite au moyen du lecteur de cartes ou de l'écran. La requête peut aussi entrer dans la composition d'un programme utilisateur écrit en langage évolué, elle fait de SOMINE une extension des langages comme Fortran et Cobol. La requête réalise non seulement l'accès à la base mais également l'interface entre les langages évolués et SOMINE.

- exemple d'interface Fortran - Somine

```
CALL INTER    (A)          (A chaîne alphanumérique)
```

- exemple d'interface Cobol - Somine

```
ENTER INTER USING      A
```

Apprendre à écrire une requête pour une personne non initiée demande très peu de temps (les automates de définition de la requête du § 3-1 sont très simples). Une interrogation initiale de la caractéristique texte contenant la structure écrite en L.D.S. permet de s'assurer de la validité des requêtes à écrire.

Par exemple, si le texte contenant la structure a comme désignation le mot STRUCTURE, la requête faisant cette interrogation s'écrira :

```
I STRUCTURE #
```

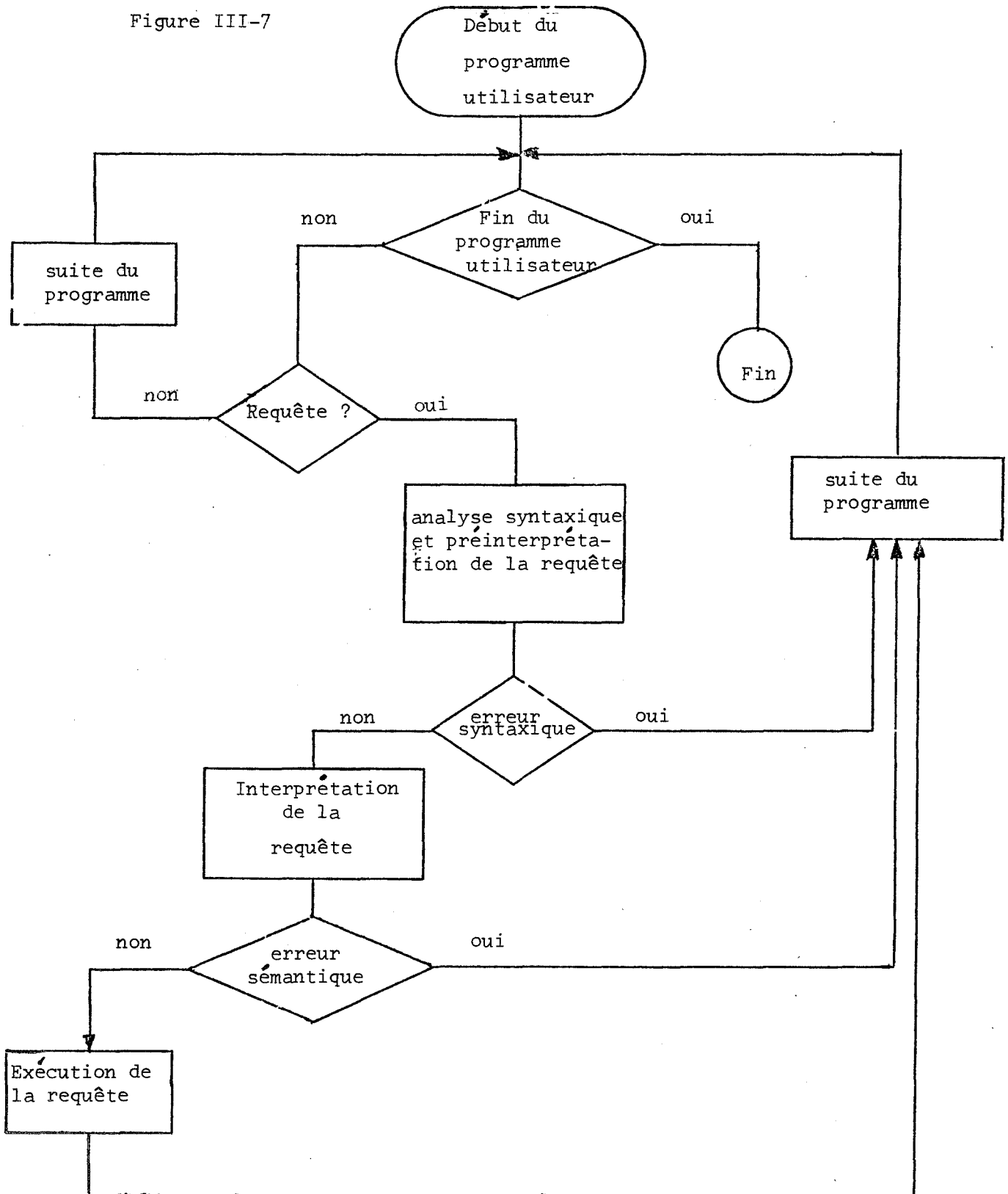
En suivant toujours notre idée d'indépendance vis-à-vis du matériel utilisé nous avons écrit tous les programmes d'accès à la base en Fortran. Le paragraphe suivant explique l'enchaînement et la réalisation des sous-programmes d'analyse, de pré-interprétation et d'interprétation d'une requête.

.../...

5 - MISE EN OEUVRE DE L'ACCES A LA BASE :

L'organigramme ci-dessous explique le déroulement d'un programme utilisateur contenant des requêtes.

Figure III-7



.../...

L'accès à la base peut se faire à l'aide de programmes standards ou de programmes définis par les utilisateurs pour une application donnée, ces programmes peuvent subir un traitement avec sortie sur l'imprimante ou sur les écrans alphanumériques.

Voici l'organigramme du programme standard utilisant les écrans (BANKECR).

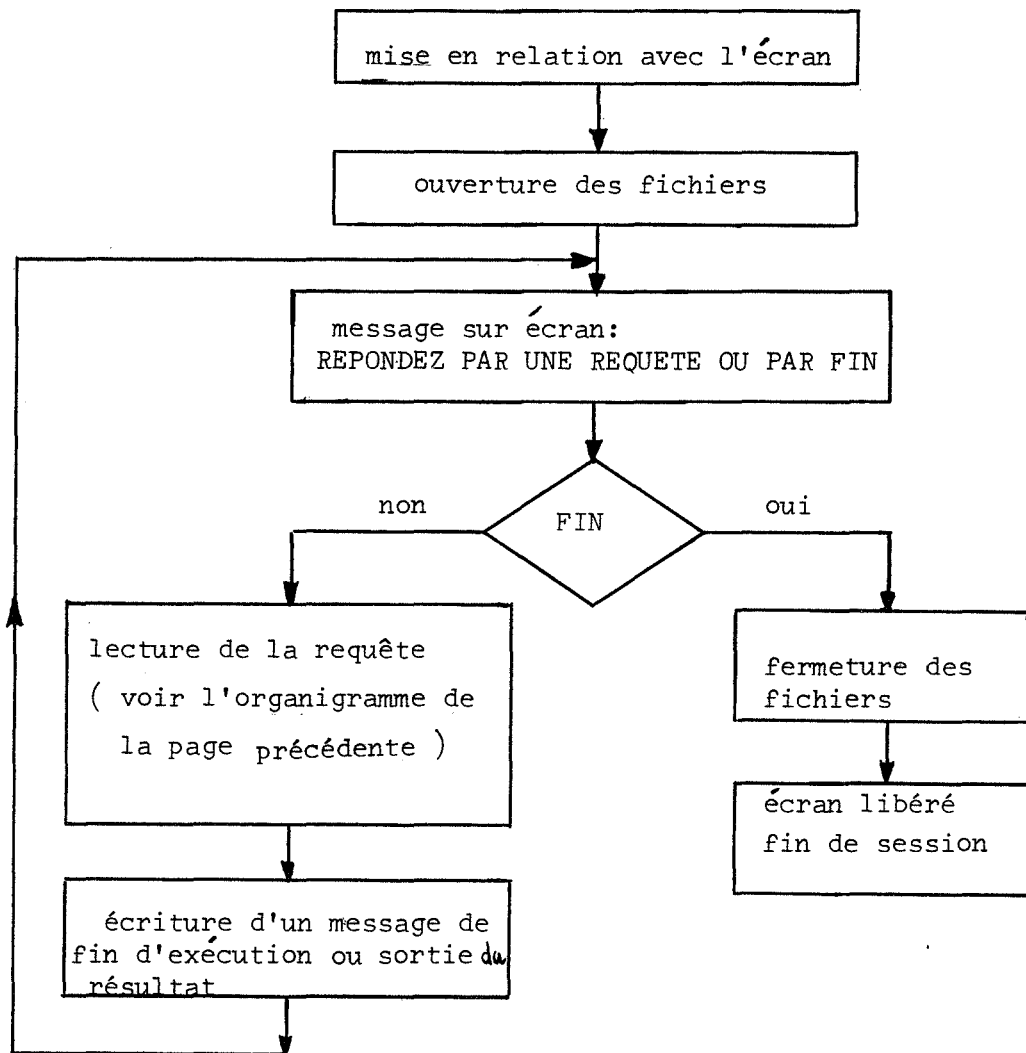


Figure III-8

Au niveau d'une des requêtes élémentaires : création, suppression, interrogation, mise à jour, le traitement des requêtes peut se résumer par le schéma ci-dessous.

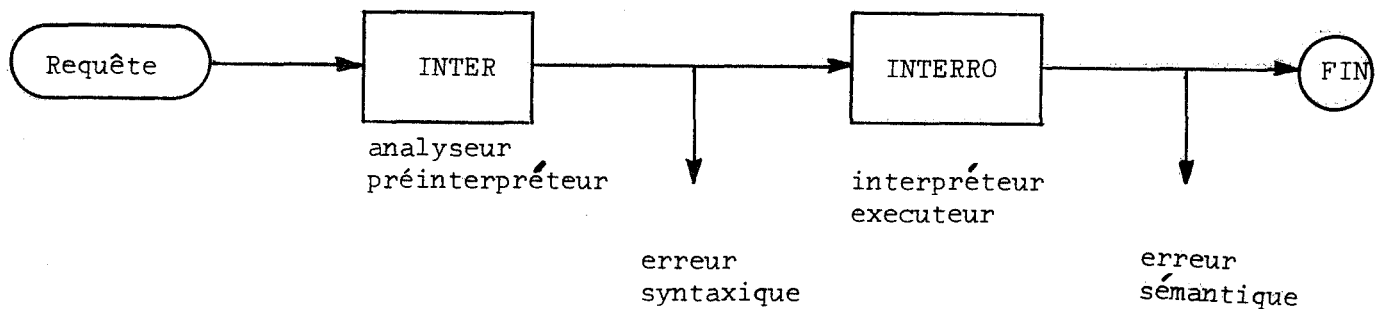


Figure III-9

5. 1. - Sous-programme d'analyse et de pré-interprétation d'une requête (INTER).

C'est le sous-programme qui est en relation avec l'utilisateur. Il a un seul argument d'entrée, c'est la constante hollerith qui représente la requête. Un certain nombre de variables sont communes au sous-programme INTER et au programme de l'utilisateur. En particulier la variable RESUL qui reste nulle si la requête n'est pas syntaxiquement correcte et la variable IAPEL qui décide de la nature de la sortie des résultats. Si cette variable est nulle, il y a une sortie normale sur l'imprimante, si cette variable vaut 1 les résultats ne sont pas sortis (l'utilisateur peut alors les formater comme il veut), si cette variable vaut 2, l'information est alors sortie sur l'écran.

5.1.1. - Les fonctions de ce sous-programme :

Elle sont multiples, nous allons énumérer les principales.

- INTER analyse syntaxiquement la requête en parcourant l'automate d'états finis du § 3-1. Cette analyse peut avoir lieu après avoir mis la requête dans un tableau à raison d'un caractère alphanumérique par mot.

- Il détermine le niveau de la citation : c'est le nombre de transitions vers l'état d'entrée de l'automate citation.
- Il code le mode de la citation
- Il empile les identificateurs rencontrés dans la requête en commençant par celui de plus bas niveau (chacun d'eux occupe quatre mots du tableau TACIT)
- Il place la valeur des démonstratifs ou les numéros de réalisation dans un tableau NUCIT. Lorsque le démonstratif ou le nombre est absent la valeur 0 est mise dans le tableau, enfin le séparateur "ayant" est codé - 1.
- Dans le cas d'une mise à jour, il garde l'information que la base doit recevoir. Suivant la nature de l'information des variables différentes sont utilisées.

La variable entière NUME reçoit les variables numériques entières.

La variable réelle RNUME reçoit les variables numériques réelles.

Dans ces deux cas, il y a transformation de la chaîne alphanumérique représentant le nombre en une **chaîne** numérique.

Lorsque la valeur est un mot ou un texte, il est mis dans le tableau INFO, la dimension de la place utilisée est donnée par la variable KINFO.

- Une autre fonction du sous-programme est d'envoyer des messages d'erreur syntaxique qui localise l'erreur dans le cas d'une requête incorrecte. Il y a alors retour au programme principal et possibilité de tester la présence de l'erreur à l'aide de la variable témoin RESUL. Ceci permet à l'utilisateur d'interrompre ou de modifier le cours de son programme s'il juge que l'erreur peut perturber le reste de son travail.

5.1.2. - Exemple :

Voici un exemple de requête de mise à jour de la caractéristique PRIX de la structure-exemple exposée dans la partie II .

M PRIX DE LA REPARATION 3 DE LA VOITURE X(5) = 100.5 #

Si X(5) à la valeur 2 par exemple, les résultats de la pré-interprétation seront les suivants :

Requête syntaxiquement correcte

Mode = mise à jour

niveau = 3

TACIT	NUCIT	RNUME
PRIX	0	100.5
REPA RATI ON	3	
VOIT URE	2	

Figure III-10

Après pré-interprétation, les variables que nous avons citées précédemment contiennent toutes les informations transmises par la requête. Ce sont elles qui vont constituer les arguments d'entrée du sous-programme d'interprétation INTERRO.

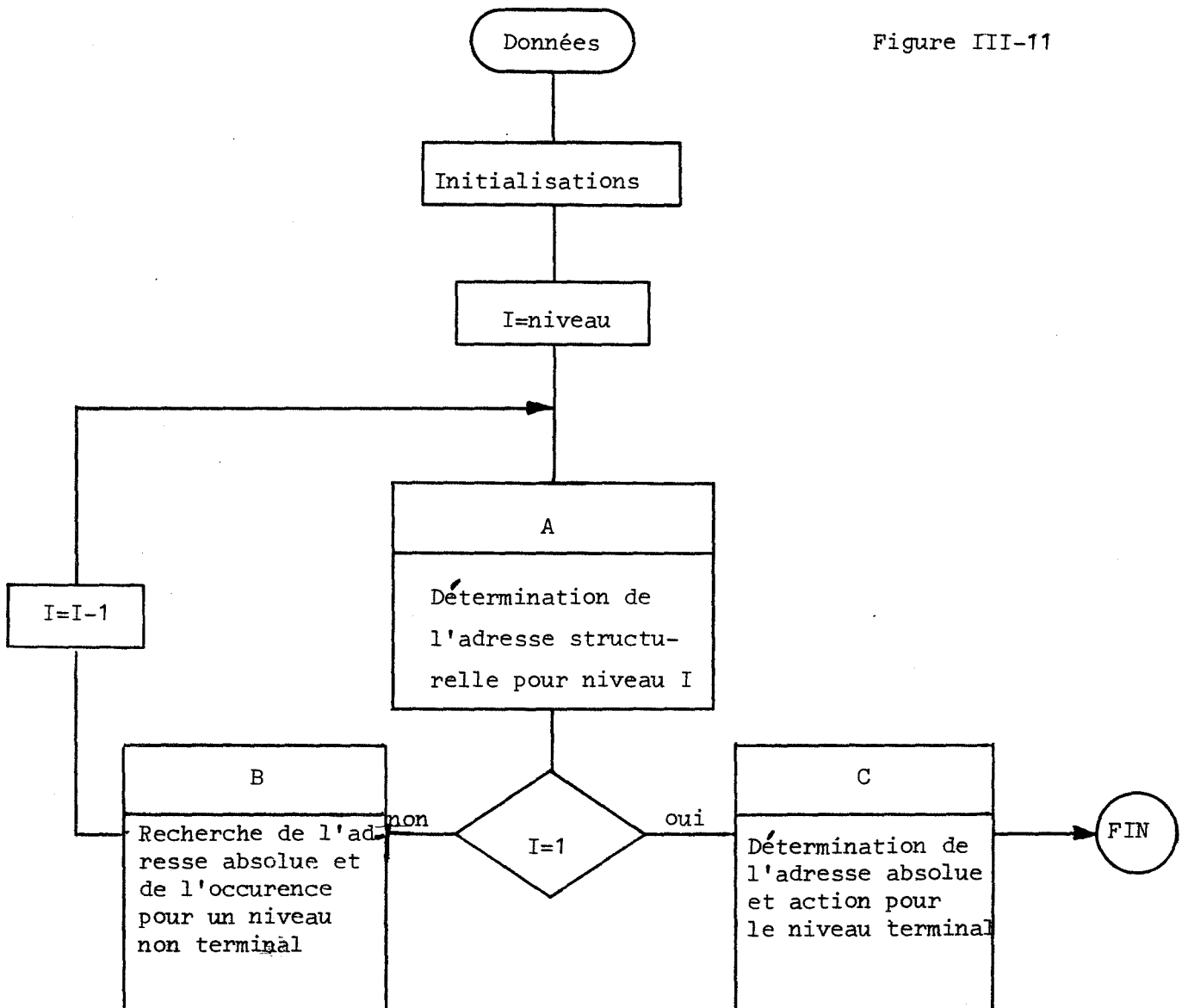
5. 2. - Sous-programme d'interprétation (INTERRO)

Les variables d'entrée vont guider le cheminement dans la structure de mémorisation interne. Dans celle-ci, la descente s'effectue par étapes, chacune d'elles assurant le passage d'un niveau au niveau inférieur. On peut distinguer deux parties dans chacune de ces étapes de recherche •

- Dans le fichier-structure, c'est l'identificateur du noeud d'arrivée de l'étape qui est recherché, on obtient ainsi son adresse structurée.

- Cette adresse structurée permet le calcul de l'adresse absolue dans le fichier-donnée de la réalisation correspondante à l'étape considérée. Dans le cas de l'étape finale, il y a en plus détermination de l'action à entreprendre et réalisation de cette action sur le fichier-données.

Dans le cas d'une requête élémentaire ne contenant pas le séparateur tout le programme d'interprétation peut être schématiser par l'organigramme suivant :



5.2.1. - Détermination de l'adresse structurelle de l'identificateur situé au niveau I d'une requête (partie A de l'organigramme de la page précédente)

La valeur de l'identificateur a été placée précédemment dans un tableau "TACIT". A chaque niveau de la requête il y a recherche de l'adresse structurelle de l'identificateur, elle se fait à partir de l'adresse structurelle du père c'est - à - dire celle de l'identificateur contenu dans TACIT au niveau immédiatement supérieur.

Cette méthode de recherche est itérative et elle est initialisée par l'adresse structurelle du fichier qui est nulle. La stratégie de recherche est différente à chaque niveau, elle dépend de la nature de la caractéristique père .

- Si $NUCIT(I) = -1$, la caractéristique de niveau immédiatement supérieur est de type inverse. L'adresse structurelle de la caractéristique citée en inverse est située dans le quatorzième mot de la sous-page relative au père /16/

- Si $NUCIT(I + 1) = -2$, la caractéristique père est de type référence. Dans le quatorzième mot de la sous-page du niveau supérieur se trouve l'adresse structurelle à partir de laquelle devra être effectuée la recherche, ensuite on est ramené au cas général exposé plus loin.

Ces deux derniers cas ne peuvent pas se produire au niveau le plus haut de la citation.

- Cas général : La recherche se fait en utilisant la fonction d'accès au premier fils et si nécessaire la fonction d'accès au premier frère de ce fils puis au frère de celui-ci.... ceci jusqu'à épuisement de la lignée. Dans le cas où le père est une entité-choix, c'est la valeur de l'occurrence qui permet de retrouver dans la représentation de la liste de valeurs-choix l'adresse structurelle du premier fils ayant cette occurrence.

Si un identificateur est écrit incorrectement ou s'il ne désigne pas une caractéristique de la structure, c'est dans cette partie du programme que l'erreur est décelée.

Si le niveau le plus bas de la citation est atteint, il y a passage à la partie C sinon c'est la partie B de l'organigramme de la page précédente qui est exécutée.

5.2.2. - Calcul de l'adresse absolue (et éventuellement de l'occurrence) de l'identificateur de niveau I de la citation (partie B de l'organigramme du § 5-2 : cas où le niveau (I) n'est pas terminal).

La recherche de l'adresse absolue pour un niveau I se fait récursivement, on ajoute à l'adresse absolue du niveau supérieur un nombre dont le calcul dépend du type de la caractéristique. Il y a initialisation

du calcul en prenant une adresse absolue nulle au début de l'interprétation. Nous allons exposer les différents cas qui peuvent se présenter. Remarquons que dans les niveaux intermédiaires d'une citation, on ne peut rencontrer que des caractéristiques ayant des descendants, c'est-à-dire les blocs, les entités et les entités-choix ainsi que les références.

- Dans le cas d'un bloc l'adresse absolue dans le fichier-données est obtenue en ajoutant à l'adresse absolue précédente l'adresse relative qui est contenue dans le grain du plexe correspondant.

- Dans le cas d'une entité ou d'une entité-choix, il faut d'abord s'assurer que la réalisation considérée a été créée, si oui, il y a recherche de son occurrence et l'adresse absolue est obtenue en ajoutant à l'ancienne adresse absolue la longueur de la chaîne de bits d'existence ainsi que la longueur de la **place** occupée par les réalisations situées avant celle dont il est question.

- Dans le cas d'une référence, il y a lecture de l'adresse absolue dans la chaîne de référence

- Dans le cas d'un inverse, il y a recherche du bit représentant la réalisation. S'il est égal à 1 la caractéristique possède la propriété filtre, sinon l'analyse est arrêtée.

A ce niveau, un certain nombre d'erreurs sémantiques peuvent être détectées et signalées. Il s'agit de réalisations d'entités non encore créées ou impossibles à créer, ou de chaînes d'inverses inexistantes. Il y a ensuite décrémentation de la variable qui indique le niveau de l'analyse et retour à la partie A.

5.2.3. - Calcul de l'adresse absolue pour le niveau terminal de la citation et exécution de l'action à entreprendre (partie C de l'organigramme du § 5-2)

A ce niveau, tous les types de caractéristiques peuvent être rencontrés. Pour celles qui ont déjà été mentionnées dans la partie B le calcul de l'adresse absolue se fait de la même façon à ce niveau. Pour les caractéristiques des autres types, c'est-à-dire celles qui interviennent dans le mode interrogation ou mise à jour le calcul de l'adresse absolue se fait en additionnant l'adresse absolue du niveau père avec l'adresse relative par rapport à ce niveau.

Nous pouvons indiquer les erreurs sémantiques qui sont décelées dans cette partie du programme. Il s'agit en particulier de l'impossibilité de créer ou de supprimer une réalisation d'entité, d'une erreur de mode ou bien de la présence d'une information à stocker dans la banque qui dépasse les limites prévues par la structure.

Les tableaux suivants indiquent les actions qui peuvent être entreprises après une requête. Nous avons énuméré tous les types de caractéristiques terminales avec les modes de requêtes qui sont susceptibles de leur être appliqués. Ces tableaux sont faits pour le cas où une sortie immédiate des résultats sur l'imprimante est demandée. C'est le cas où la variable IAPEL prend la valeur 0, rappelons que si cette variable prend la valeur 1, le formatage des résultats est laissé au soin de l'utilisateur et si elle prend la valeur 2 le résultat de l'interrogation est écrit sur un écran alphanumérique, le format de sortie est alors quatre caractères alphanumériques par mot.

TYPE DE LA CARACTE-
RISTIQUE TERMINALE

INTERROGATION

MISE A JOUR

MOT	Lecture dans le fichier-données à l'adresse absolue AABS et sur une longueur donnée par la taille de la caractéristique . Sortie du résultat en format A (Fortran).	Ecrire dans le fichier-données à l'adresse AABS le contenu du tableau INFO qui contient le mot.
TEXTE	"	Ecrire dans le fichier-données à l'adresse AABS le contenu du tableau INFO et sur la longueur égale à KINFO
NUMERIQUE E	Lecture dans le fichier-données du mot situé à l'adresse AABS. Sortie du résultat en format I (Fortran)	Ecrire dans le fichier-données à l'adresse AABS le contenu de la variable NUME
NUMERIQUE R D	Lecture dans le fichier-données du mot d'adresse AABS Sortie en format F ou E Sortie en format D	Ecrire dans le fichier-données à l'adresse AABS le contenu de la variable RNUME ainsi que celui de la variable RNUME1.
Liste de valeurs Liste de valeurs - choix	Lecture dans le fichier-données à l'adresse absolue AABS du numéro de la valeur de la liste. Lecture dans le fichier structure de la désignation correspondante. Sortie en format A	Recherche dans le fichier-structure du numéro de la valeur de la liste. Ecriture de ce numéro dans le fichier-données à l'adresse AABS

type de la caractéristique terminale

interrogation

mise à jour

IDEM	Le résultat de cette interrogation est le même que celui de la caractéristique citée par l'idem	Même remarque
INVERSE	Recherche du bit ayant le numéro considéré dans la chaîne de bits située à l'adresse AABS. Si ce bit a la valeur 1, la caractéristique à la propriété inverse	Mise a 1 du bit de la chaîne d'existence ayant le numéro considéré. Cette chaîne est située à l'adresse que l'on a calculée : AABS.

type de la caractéristique terminale

création

suppression

ENTITE ENTITE-CHOIX	Il y a recherche du bit de la chaîne d'existence dont le numéro d'ordre est le numéro de réalisation ensuite, la valeur 1 est attribuée à ce bit	Il'y a mise à 0 du bit d'existence correspondant et modification éventuelle de la chaîne des références
REFERENCE	Il'y a ajout d'un maillon à la chaîne des références située à l'adresse AABS.	Il y a mise hors circuit d'un maillon de la chaîne des références.

6 - APPLICATIONS :6. 1 - Exemple de programme utilisateur :

Le programme suivant constitue une illustration des possibilités des requêtes élémentaires et il se rapporte à l'exemple test décrit en II - 9

```

1      DEFINE FILE 50(99,257,II,IV),60(99,257,II,IV)
2      INTEGER COMPARE,X
3      COMMON X(20),RNUMF,NUMF,INFO(320),KINFO,IFCR(16,20),RESUL,IAPFL
4      CALL FORMFICH(2,97,4)
5      CALL INFICH(3)
6      IAPFL=0
7      DD1I=1,20
8      1   Y(1)=0
9          CALL STRTCLK
10         DD2I=1,10
11         Y(1)=1
12         CALL INTER('O PERSONNE X(1) # ')
13         2   CALL INTER('O VOITURE X(1) # ')
14         CALL PRNTCLK
15         CALL STRTCLK
16         DD3I=1,10
17         DD3I=1,20
18         X(1)=I
19         Y(2)=J
20         3   CALL INTER('O REPARATIONS X(2) DE LA VOITURE X(1) # ')
21         CALL PRNTCLK
22         CALL STRTCLK
23         CALL INTER(' M JOUR DE DATE = 20 # ')
24         CALL INTER(' M MOIS DE DATE = FEVRIER # ')
25         CALL INTER(' M AN DE DATE = 1975 # ')
26         CALL INTER(' M SEXE DE LA PERSONNE 1 = MASCULIN # ')
27         CALL INTER(' M NOM DE LA PERSONNE 1 = DUPAN # ')
28         CALL INTER(' M NATURE DE LA REPARATION 1 DE LA VOITURE 1
29         A= TOIERIE # ')
30         CALL INTER(' M PRIX DE LA REPARATION 1 DE LA VOITURE 1 =
31         71000.0 # ')
32         CALL INTER(' I PRIX DE LA REPARATION 1 DE LA VOITURE 1 # ')
33         IF(RNUMF=100.)20,10,10
34         10  CALL INTER(' I NATURE DE LA REPARATION 1 DE LA VOITURE 1 # ')
35         IF(RESUL)20,20,11
36         11  CALL INTER(' I AN DE DATE # ')
37         CALL INTER(' I MOIS DE DATE # ')
38         20  CALL PRNTCLK
39         CALL OUTFICH
40         STOP
41         END
NUMBER OF DIAGNOSTIC MESSAGES IS 0

```

.../...

6. 2 - Résultats obtenus :

La requête présentée à la ligne 20 est une requête de création d'une réalisation d'entité, son exécution provoque la sortie d'un message indiquant le numéro de la réalisation qui vient d'être créée ainsi que le conseil de mise à jour des caractéristiques filles en commençant par la liste de valeurs s'il s'agit d'une entité choix.

Les requêtes de mise à jour des lignes 23 et 30 donnent un message indiquant que la mise à jour a pu être exécutée.

Les requêtes d'interrogation des lignes 32 à 37 provoquent la sortie des valeurs mises à jour précédemment à savoir :

1000.0
TOLERIE
1975
FEVRIER

6.3 - Remarques :

Le programme précédent suggère les remarques suivantes :

- dans la version actuelle de SOMINE, il faut définir les deux fichiers (structure et données) qui ont les clés 50 et 60. La définition se fait à l'aide de l'instruction DEFINE FILE dans laquelle le nombre 257 indique la longueur de l'enregistrement physique (ligne de programme n° 1).

- l'instruction :

CALL FORMFICH (2, 97, 4)

est obligatoire lors de la toute première utilisation du fichier-données (ligne n° 4). Cette instruction appelle une sousroutine qui a pour but de formater le fichier 2. Dans l'exemple précédent ce fichier aura 97 pages composés de 2⁴ sous-pages.

- le sous-programme INFICH (ligne n° 5) réalise l'ouverture des deux fichiers nécessaires à la base.

- l'instruction :

CALL OUTFICH

effectue la fermeture des fichiers et en particulier sauvegarde les dernières

modifications apportées à la base et encore contenues dans le plan primaire.

- ne pas oublier également l'instruction COMMON (ligne n°3) car les variables mises en commun entre les différents programmes servent de véhicules à l'information qui doit circuler entre la base et les organes de sortie du calculateur.

- il est plus prudent de remettre à ZERO le tableau des démonstratifs avant une utilisation.

- l'utilisation des boucles DO (ligne n° 10) et même des boucles DO imbriquées (ligne n° 16) permet d'exécuter plusieurs interrogations à l'aide d'une seule requête, il suffit pour cela de faire varier les valeurs des démonstratifs qui y figurent.

- nous remarquons qu'à tout moment un test peut être effectué sur la valeur de la variable RESUL et qu'il peut orienter la poursuite du programme (ligne n° 35)

- enfin en faisant varier la valeur de IAPEL (ligne n° 6) on peut diversifier les sorties de résultats.

L'exemple du § 6-1 est un cas de programme dans lequel les requêtes sont perforées sur des cartes, mises à l'intérieur du programme, et exécutées avec lui.

6. 4 - Exemples de programmes standards :

a) Le programme qui suit permet de faire ,exécuter des séries de requêtes sur une base de type SOMINE .

Il peut être mis sur une bibliothèque disque et nous verrons dans la partie IV que l'on peut également en déclencher l'exécution à partir de la base elle-même .

```

DEFINE FILE 50 (99, 257, U, IV), 60 ( 99, 257, U, IV)
DATA A / 40 ( 4 H          ) /
INTEGRER X , A (40)
COMMON X (20), RNUM, NUME, INFO (320), KINFO, IECR (16, 20), RESUL, IAPEL
CALL INFICH (2, 97, 4 )
3 READ (1.,1 ,END= 2 ) A
1 FORMAT (20 A 4 / )

```

.../...

```

CALL INTER ( A)
GOTO 3
2 CALL OUTFICH
STOP
END

```

b) Il existe également un programme standard qui utilise l'écran, il s'appelle BANKECR.

Son exécution fait apparaître sur l'écran le message suivant :

REPONDEZ PAR UNE REQUETE OU PAR NON #

sous réserve d'avoir mis la variable IAPEL à 2 les sorties des requêtes, les messages de bonne exécution ou ceux d'erreurs apparaissent sur l'écran.

Les sessions de travail à l'écran sont très agréables car il s'agit d'un véritable dialogue entre l'utilisateur et la base. De plus, pour les requêtes de mise à jour qui portent sur des textes longs la rédaction et la correction des fautes est sont faciles à faire avant la transmission du contenu de l'écran à la base.

Cette procédure présente cependant l'inconvénient d'être légèrement plus longue que l'accès par cartes mais on comprendra tout son intérêt dans la thèse de M. GAILLARD /15/ qui traite de l'utilisation de SOMINE dans un but pédagogique.

6. 5 - Remarques relatives à la programmation :

Le système SOMINE tel que nous l'avons décrit depuis le début de ce rapport représente :

- 14 Koctets pour la gestion de mémoire
- 22 Koctets pour l'implémentation de la structure
- 19 Koctets pour l'interprétation des requêtes

Quelque soit la phase du travail, mise au point de la structure ou utilisation d'une base déjà existante l'encombrement mémoire est assez important.

Le moniteur du P1175 de l'Ecole des Mines sur lequel a été

réalisé SOMINE fonctionne en multiprogrammation. La prise en compte d'un travail dépend de sa priorité et de son encombrement, il faut en effet lorsque plusieurs programmes sont en cours d'exécution attendre qu'une place suffisante soit libérée par l'achèvement d'autres travaux.

Cette contrainte a été une des raisons qui ont fait que nous avons essayé de réaliser une programmation par module. Ces modules ne sont pas tous indépendants les uns des autres. La relation qui lie deux de ces modules est la relation "appelle".

par exemple : BANKECR appelle INTER

Le langage FORTRAN interdisant la récursivité (si A appelle B, B ne peut pas appeler A) on peut représenter les programmes composant la base par une structure arborescente.

exemple :

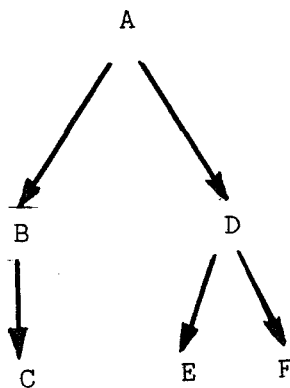


Figure III-12

Pour éviter l'encombrement mémoire et par la même accélérer l'exécution des programmes, nous avons utilisé la technique de l'overlay. Nous allons brièvement l'expliquer à l'aide de l'exemple précédent. A un moment de l'exécution le programme principal A doit être en mémoire centrale ainsi qu'un sous-arbre partant de A. Par exemple si on exécute B le programme F n'a aucune raison d'être en mémoire centrale puisqu'il n'est pas appelé par B : ni par aucun sous-programme descendant de B.

On peut donc amener en mémoire centrale le ou les programmes nécessaires à un niveau donné de la structure de tous les programmes. Pour chaque niveau le choix est guidé par des expressions booléennes qui figurent sur les cartes d'overlay.

.../...

exemple :

expression booléenne

représentation sur les
cartes d'overlay

A
B ou D
C ou (E et F)

A
B, D
C, (E + F)

Naturellement dans une expression comme B, D c'est la taille du plus grand des deux programmes qui est réservée.

Nous arrivons par cette méthode à une occupation mémoire d'environ 80 K octets pour une utilisation de SOMINE en traitement par lots ou à l'écran.

7 - AMELIORATIONS APORTEES A L'INTERPRETEUR DE BASE :

=====

Nous avons exposé dans les paragraphes précédents l'implémentation et l'utilisation des quatre requêtes de base : INTERROGATION, MISE A JOUR, CREATION, SUPPRESSION qui permettent la vie des informations dans une base. Nous avons également mentionné le mode FREQUENCE qui permet de mesurer le nombre d'accès à une caractéristique d'une base Somine.

Ces 5 modes de requêtes ont permis de faire subir, aux collections d'informations mises en jeu dans les recherches d'application du projet SOMINE, l'ensemble des traitements qui étaient nécessaires.

C'est cependant lors de la mise en oeuvre des applications du système SOMINE qu'il nous a semblé souhaitable d'apporter quelques améliorations à l'outil de base ceci afin d'avoir une utilisation plus performante. Les améliorations peuvent être de deux types : le temps d'accès à la base pour un travail donné peut être raccourci ou bien la formulation de cet accès peut être simplifiée.

.../...

Nous avons surtout pensé à six améliorations :

- * les requêtes groupées
- * les requêtes booléennes
- * la synonymie
- * les requêtes à valeur variable
- * les requêtes contenant le séparateur "tout"
- * les requêtes discriminantes.

7. 1 - Les requêtes groupées :

Il s'agit de pouvoir interroger ou mettre à jour toutes les caractéristiques filles d'une réalisation d'entité à l'aide d'une seule requête.

- le résultat d'une telle requête

Dans le cas de l'interrogation le résultat de la requête est la sortie de la liste des désignations des caractéristiques filles ainsi que celle des valeurs qui leurs sont associées et qui sont contenues dans la base au moment de l'accès.

Dans le cas de la mise à jour deux cas sont à considérer suivant qu'il s'agit d'un accès en batch ou en conversationnel (écran).

En batch :

pour éviter de devoir interdire un nouveau séparateur (il devrait être placé entre chaque valeur à attribuer à la liste des désignations des caractéristiques à pourvoir) nous avons pensé qu'il valait mieux que ces valeurs soient lues sur des cartes de données du programme d'utilisation, ainsi elles peuvent plus aisément être changées sans modification du programme d'accès. Les valeurs données seront attribuées au premier fils de l'entité en question puis au frère de celui-ci.... jusqu'au dernier fils, c'est-à-dire suivant la méthode de cheminement dans l'arbre structure qui a été exposée au § 2.3.

En conversationnel :

le programme envoie la désignation des caractéristiques à pourvoir et l'utilisateur renvoie la valeur à leur attribuer.

- La forme d'une telle requête :

Dans l'automate de la requête l'identificateur désignant l'élément terminal est remplacé par le mot "TOUT" suivi de la citation d'entité correspondante.

L'automate peut être complété ainsi:

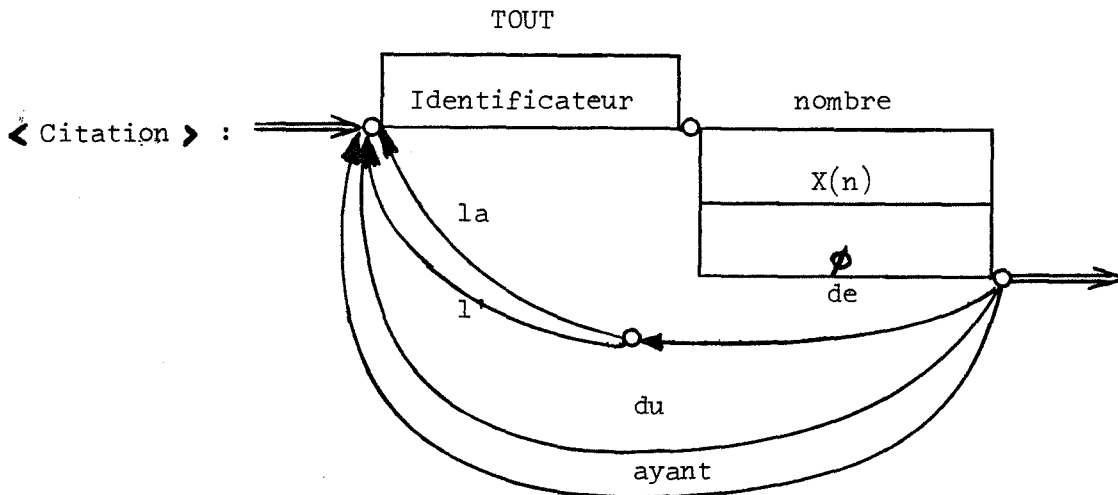


Figure III-13

- la mise en oeuvre d'une telle requête :

Les modifications de programmation sont très légères puisque c'est toujours le même automate qui permet de décrire la requête. En effet, nous avons seulement rajouté une transition au niveau de la désignation de l'élément terminal.

7. 2 - Les requêtes booléennes :

Il s'agit de faire une meilleure utilisation des chaînes d'inverses associées aux caractéristiques de type inverse.

Une caractéristique peut être filtrée par une caractéristique inverse, ceci correspond à la création d'une chaîne de bits, les numéros des bits dont la valeur est zéro correspondent aux numéros de réalisation de l'entité qui n'ont pas la propriété désignée par la caractéristique inverse, ceux dont la valeur est un possèdent la propriété.

Nous avons pensé compliquer le filtre en permettant les expressions booléennes de caractéristiques inverses, ainsi sont créés des filtres plus puissants ou plus fins suivant que les éléments de l'expression

booléenne sont souvent vérifiés en même temps ou rarement.

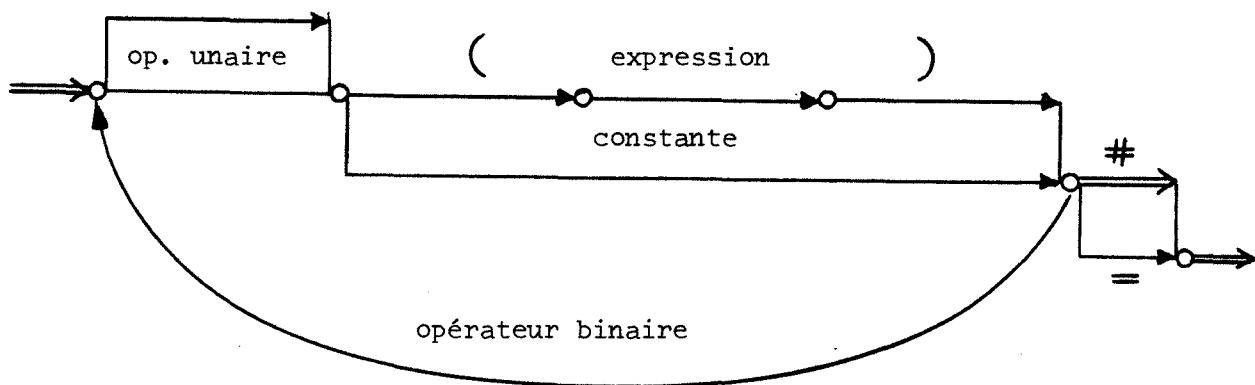
- le résultat d'une telle requête :

Une telle requête permet de savoir si une réalisation d'entité possède une propriété composée ou sa négation.

Cette requête permet les interrogations et les mises à jour conditionnelles, dans le cas de l'interrogation le résultat est "vrai" ou "faux" dans le cas de la mise à jour celle-ci n'est faite que si le résultat de l'expression booléenne est "vrai".

- la forme de la requête :

Voici l'automate de l'expression booléenne.



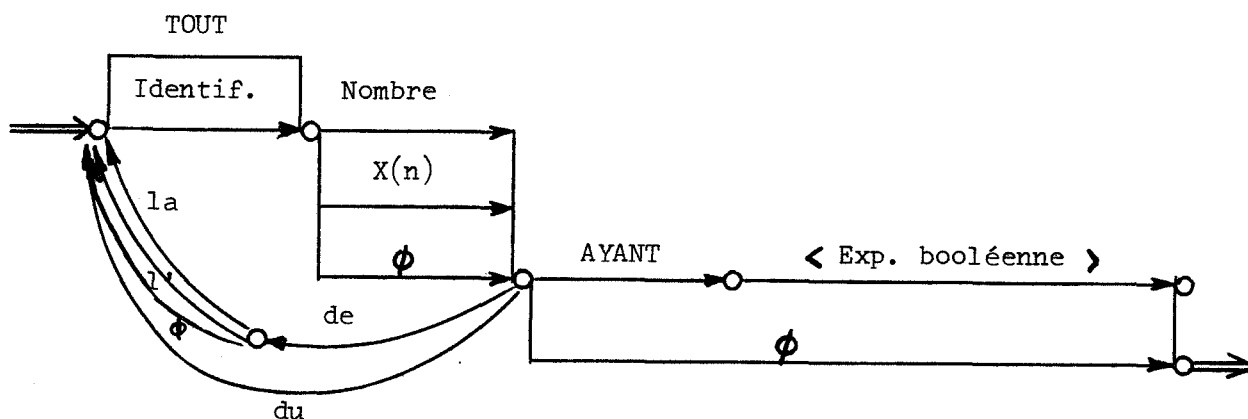
Opérateur unaire : négation

Opérateur binaire : ET , OU

Figure III-14

L'automate de la citation hiérarchique peut être complété comme indiqué ci-dessous :

Figure III-15



- la mise en oeuvre de la requête :

L'automate précédent montre qu'il suffit de faire accepter par l'interpréteur des requêtes l'existence et l'analyse des expressions booléennes. Pour cela nous avons ajouté, un module supplémentaire à l'interpréteur, il s'agit du sous-programme BOOLE qui s'exécute lorsque le séparateur rencontré est "AYANT".

Ce programme prend en entrée l'expression et l'analyse syntaxiquement et sémantiquement en utilisant l'automate de l'expression, la structure des informations et le fichier des données, en sortie il restitue une variable logique qui indique si la réalisation d'entité considérée a ou non la propriété cherchée et une variable entière qui permet de tester si la requête a pu être menée à bien.

7. 3 - La synonymie :

Permet d'éviter d'avoir à réécrire une partie d'une citation hiérarchique ayant pour point d'arrivée la racine de l'arbre structure.

Par exemple si :

Y = REPARATION 1 DE LA VOITURE 1

on pourra écrire : I NATURE DE Y #

Pour la mise en oeuvre de cette simplification dans l'écriture d'une requête fréquemment utilisée on peut penser à plusieurs solutions. La plus simple étant peut-être de stocker la chaîne alphanumérique qui représente la citation dans un tableau auquel on associe un identificateur et les résultats de la préinterprétation de cette partie de la requête.

7. 4 - Requêtes à valeur variable :

Lors d'une requête d'interrogation le résultat (la valeur) est mis automatiquement par le programme d'interprétation dans une variable qui correspond au type (numérique réel ou entier, texte....) de la caractéristique interrogée. Ceci est particulièrement intéressant car ce résultat peut subir (en aval de cet accès) dans le programme utilisateur tous les traitements possibles (calculs comparaisons...)

La requête à valeur variable aura pour but de permettre l'action duale de la précédente à savoir la mise à jour d'une caractéristique avec le contenu d'une variable pouvant avoir subi des traitements en amont de cet accès à la base.

Cette requête présente plusieurs avantages outre celui qui vient d'être cité, les requêtes de mise à jour pourront avoir une forme plus banalisée et en particulier on pourra faire de la mise à jour en série à l'intérieur d'une boucle (DO FORTRAN par exemple), seule la valeur de la variable changera, celle-ci pouvant être lue au fur et à mesure des mises à jour. L'exemple suivant indique une forme possible d'une telle requête.

```
exemple : DO 5 I = 1,10
           READ (1,15) ( INFO (I), I = 1,4)
15        FORMAT ( 4A4)
           CALL INTER ( 'M NOM DE LA PERSONNE X(I) = INFO #' )
10        CONTINUE
```

Les formes possibles pour la requête :

La modification intervient dans l'automate valeur(figure III.6)

Celui-ci peut-être complété de la façon suivante :

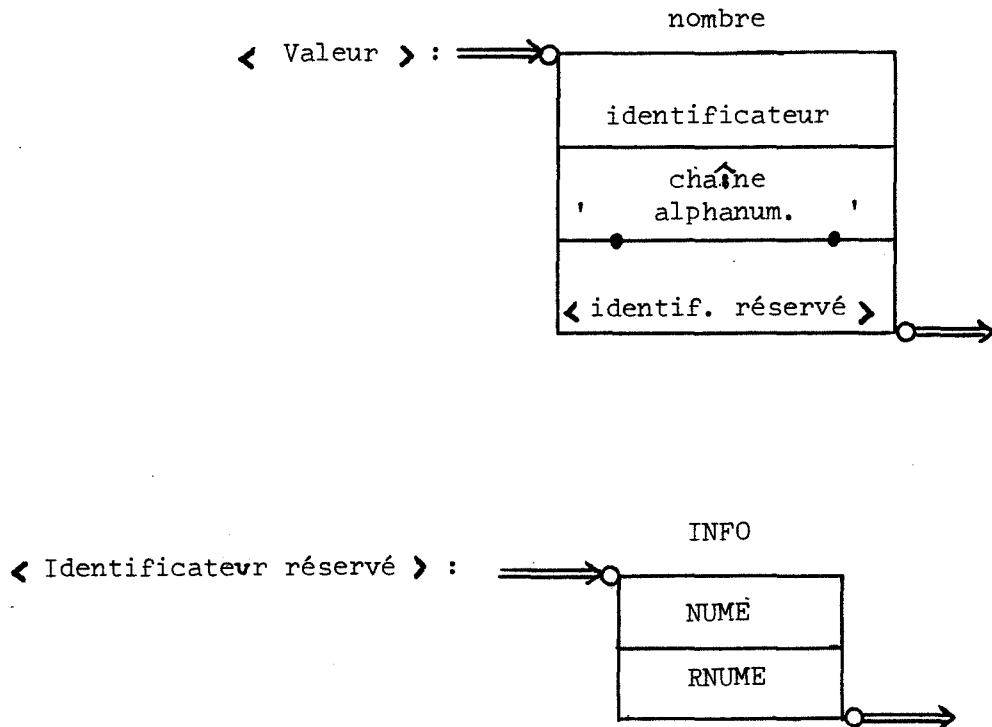


Figure III-16

La mise en oeuvre de la requête :

Lors d'une requête de mise à jour après le signe " = " il y a recherche des identificateurs réservés. Si un tel identificateur est rencontré c'est son contenu qui est transféré dans la base à l'adresse qui a été calculée par le programme d'interprétation. Cette procédure facilite les mises à jour et réduit le temps d'interprétation, en particulier pour les valeurs numériques qui ne subissent pas la transformation de chaîne alphanumérique en numérique.

7. 5 - Les requêtes contenant le séparateur "TOUT".

Des modifications ont été faites pour pouvoir analyser les citations contenant le séparateur "TOUT", elles entraînent la gestion de plusieurs tableaux car il faut retenir les adresses structurelles et absolues pour tous les niveaux de la citation afin de ne pas refaire l'interprétation un grand nombre de fois. Cette procédure allonge le temps d'accès à la banque pour les requêtes ne contenant pas ce séparateur.

Nous avons écrit un nouveau programme qui est une généralisation du précédent, c'est lui qui effectuera l'interprétation dans le cas où l'analyse syntaxique aura décelé la présence d'un "tout".

7. 6 - Requêtes discriminantes :

Les applications du S.G.B.D. SOMINE /15/, /16/, que nous avons mises en oeuvre ont montré que les requêtes précédemment définies suffisaient à résoudre les problèmes envisagés. Cependant dans le cas d'une application de SOMINE dans le domaine de la gestion une requête "discriminante" serait sans doute utile et peut-être indispensable. Par requête "discriminante" nous voulons désigner la possibilité laissée à l'utilisateur de faire des interrogations ou des mises à jour conditionnelles, la condition portant sur une valeur particulière d'une caractéristique simple. Par exemple en se référant toujours à l'exemple -test de la partie II :

" interroger la couleur des voitures dont la marque est renault "

Ce problème est déjà partiellement résolu par les requêtes booléennes dans le cas où la caractéristique discriminante est une variable ou une expression booléenne.

Dans le cas général, on peut envisager la création d'une caractéristique de type "discriminante".

Une solution consisterait à réserver une place dans le fichier données pour la liste des valeurs de la caractéristique discriminante et lors de l'interrogation la recherche se ferait en parcourant la liste précédente.

7. 7 - Schéma général des accès possibles aux bases SOMINE :

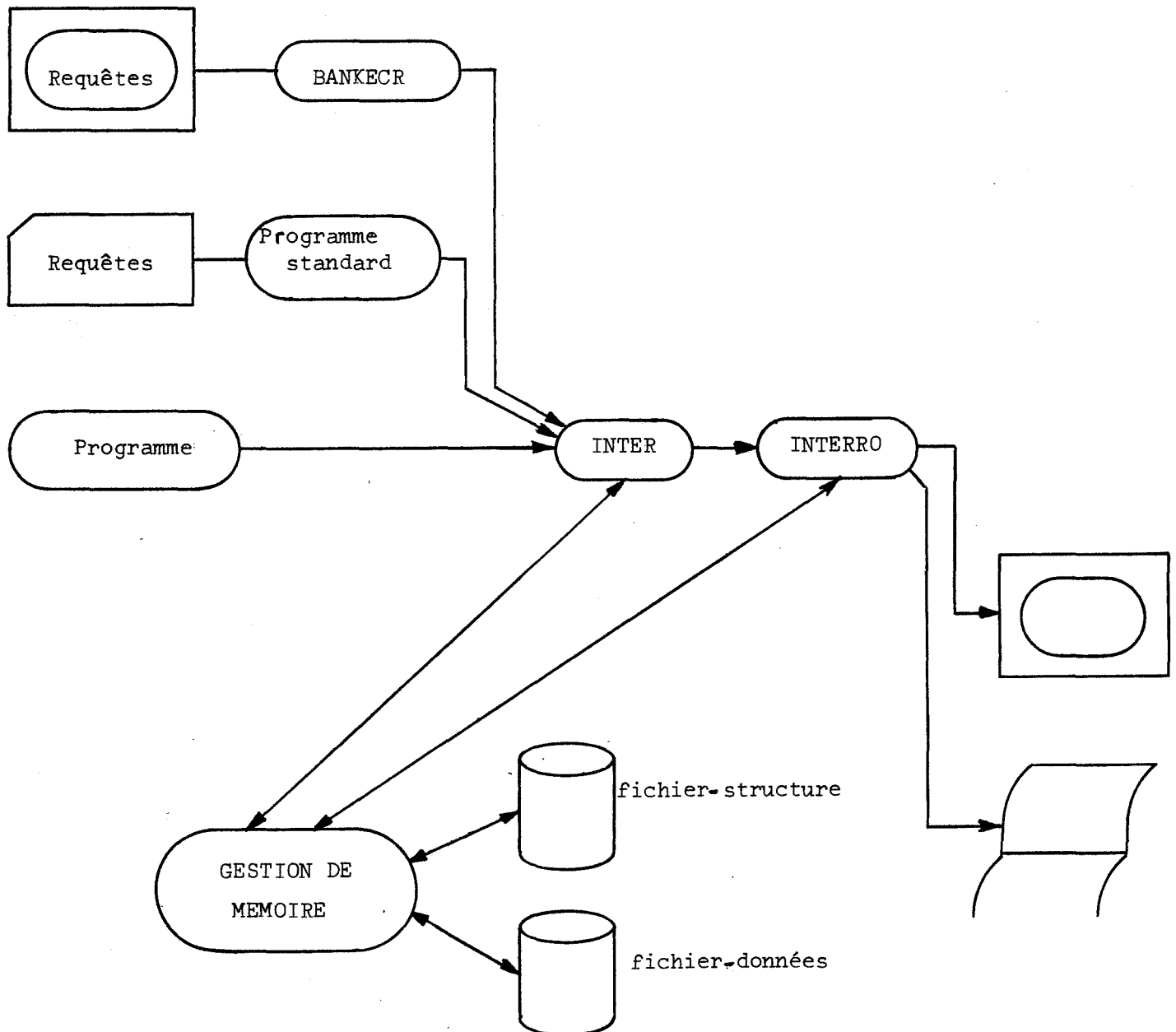


Figure III-17

8 - MULTI ACCES A LA BASE :

C'est dans cette direction que les travaux sont en cours. Monsieur THOUROUDE (étudiant à l'Ecole Nationale Supérieure des Mines de Saint-Etienne) réalise un système de temps partagé qui permettra l'accès simultané de plusieurs utilisateurs à une même base SOMINE. Cette partie est exposée dans son rapport de DEA /17/

9 - PROBLEMES LIES A LA PROTECTION DES UTILISATEURS :

Le multi-accès à une base pose le problème de la protection de cette base. Un moyen simple de le résoudre est le mot de passe permettant l'ouverture des fichiers mais il semble que le programme utilisateur puisse aussi faire sa propre protection. Nous retrouvons ainsi une des qualités de SOMINE qui est l'adaptabilité au besoin des utilisateurs. Nous pensons que des protections sont possibles mais qu'il appartient à chaque application de juger du degré et de la nature de la protection dont elle a besoin.

10 - CONCLUSION :

Comme nous l'avons souhaité la requête de base de SOMINE possède les qualités annoncées : simplicité, souplesse, et rapidité d'apprentissage. La modularité de la programmation et la simplicité des automates de définition ont permis d'apporter à l'interpréteur de base des améliorations qui augmentent les performances de la base. Ces modifications n'ont pas entraîné de grosses modifications de programmation ce qui permet de vérifier la possibilité d'évolution suivant les besoins du produit SOMINE. Plusieurs points ont été soulevés (séparateur TOUT, fiabilité...) et seule une expérimentation plus complète permettra de donner une réponse plus précise.

Nous pensons cependant que ce que nous avons déjà réalisé permettra de juger et d'évaluer notre travail.

Notre travail s'est ensuite orienté vers des applications de SOMINE à la C.A.O., l' E.A.O. et la gestion. Ce sont ces applications qui outre leur valeur d'exemple nous ont permis une réflexion plus approfondie sur l'outil que nous avons créé et ont donné l'idée de perfectionnements ou de mises au point.

Nous avons apporté une solution aux problèmes rencontrés lors de la conception et de la réalisation d'un système de base de données, il nous

reste la phase d'exploitation avec les nombreux problèmes qu'elle pose.

La base remplit les fonctions de description des données de manipulation et d'utilisation de ces données.

C'est lors de l'exploitation de notre système que les fonctions de sauvegarde, protection et sécurité vont devenir nécessaires.

Les travaux actuels portent sur les multiaccès à un système de bases de données SOMINE. Cette amélioration fera du logiciel SOMINE un produit susceptible d'intéresser des groupes de chercheurs en cours de conception d'un système général devant contenir une base de données.

Pour notre part nous avons voulu expérimenter comment une base SOMINE peut s'intégrer dans un processus plus vaste. Nous avons donc essayé de construire autour d'une base SOMINE (au besoin en la façonnant dans ce but) un système mettant en évidence et utilisant les aspects originaux de SOMINE.

Dans cette thèse va être maintenant exposé l'application à l'aide à la conception assistée par ordinateur des structures métalliques dans le but de leur assurer une tenue en service convenable.

Le lecteur trouvera principalement deux aspects :

* Un aspect informatique : SOMINE résoud-il ce problème ? améliorations apportées à SOMINE dans ce but.

* Un aspect raisonnement informatique appliqué aux sciences expérimentales. Ceci dans le but d'informatiser un domaine de recherche et d'application qui ne faisait pas encore appel à ces méthodes.

La lecture des ouvrages dont la liste suit a servi à l'élaboration et à la rédaction de cette partie du rapport .

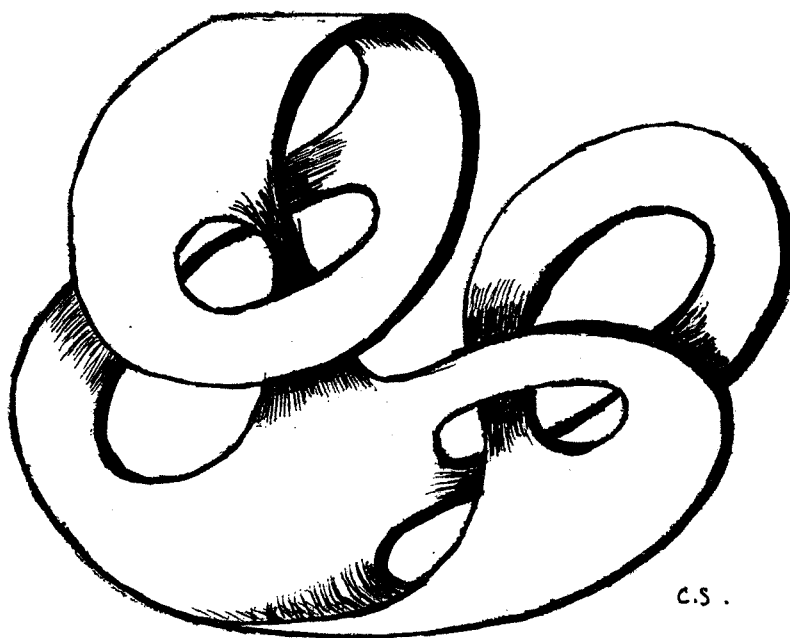
DOCUMENTATION

- /1/ - J.R. ABRIAL, J. BAS, G. BEAUME, G. HENNERON, R. MORIN, G. VIGLIANO, université de Grenoble, Institut de mathématiques Appliquées :
" projet SOCRATE : spécifications générales" - Août 1970
- /2/ - J.R. ABRIAL, J.P. COHEN, J.C. FAVRE, D. PORTAL, G. MAZARE, R.MORIN , université de Grenoble, Institut de Mathématiques Appliquées :
" Projet SOCRATE : nouvelles spécifications (version 3)" septembre 1972
- /3/ - R. THOMAS, chef du centre de calcul de l'Ecole Nationale Supérieure des Mines de St.Etienne, " évaluation du système SOCRATE en vue de son utilisation éventuelle comme support d'une banque de données sur les enquêtes faites par le Ministère du Développement Scientifique et Industriel" - décembre 1972
- /4/ - PHILIPS - " document sur le système de base de données PRINSYS CHARM"
- /5/ - Robert K. LINDSAY, University of Michigan, "In defense of Ad Hoc Systems" (computer Models of thought and language) - octobre 1972
- /6/ - G. SALTON - U.S.A. " The Smart Retrieval System" (experiments in automatic document processing) - 1971
- /7/ - F.R.A. HOPGOOD , " techniques de compilation " - DUNOD
- /8/ - J. STERN, P. LEPETIT, J.M. CHABANAS, "Initiation pratique à l'informatique" - DUNOD
- /9/ - J.C. DERMIANE, Université de Nancy I, projet CIVA " Un système de programmation modulaire" thèse d'Etat - janvier 1974
- /10/ - CHAPUIS, Université de LYON I : " Réalisation d'un analyseur syntaxique du langage COBOL fonctionnant en temps partagé." (avril 1973)
- /11/ - R. MAHL, Ecole Nationale Supérieure des Mines de St.Etienne : "cours d'informatique software : algorithme et structures de données ".
- /12/ - PAIR, professeur à l'Ecole Nationale Supérieure de la Métallurgie et de l'Industrie des Mines de NANCY, cours de structures de données et algorithmes fondamentaux - 1974

- /13/ - A. SARAZIN, Université de LYON I, "créativité et innovation"
janvier 1971
- /14/ - Datamation - Data Base Management - vol. 20 - n°9 septembre 1974
- /15/ - M. GAILLARD, thèse de doctorat de troisième cycle ,Ecole Nationale Supérieure des Mines de Saint-Etienne, " Contribution à la conception, à la réalisation et à l'utilisation du système de gestion de bases de données SOMINE : gestion des mémoires -Enseignement Assisté par Ordinateur ." 1976
- /16/ - P. MARTY, thèse de doctorat de troisième cycle , Ecole Nationale Supérieure des Mines de Saint-Etienne, " Contribution à la conception, à la réalisation et à l'utilisation du système de gestion de bases de données SOMINE : Structuration des informations ." 1976
- /17/ - G. THOUROUDE, rapport de D.E.A. - Ecole Nationale Supérieure des Mines de St. Etienne, Juin 1975
- /18/ - J. J. GIRARDOT, "Utilisation des écrans sous SYSTDS " info info de l'Ecole Nationale Supérieure des Mines de St.Etienne,, Février, Mars 1973
- /19/ -"Etude sur les software de gestion et d'interrogation des fichiers"
Délégation à l'informatique -Groupe systèmes d'informations .
(avril 1973)
- /20/ -Colloques I.R.I.A. " Banques de données " Aix en Provence, 1973

PARTIE IV

SOMINE ET L'AIDE A LA CONCEPTION ASSISTEE PAR ORDINATEUR DES PROJETS SCIENTIFIQUES ET TECHNIQUES



La créativité est " la perception d'une situation ou d'une idée sur deux plans qui sont habituellement incompatibles ..." A.Koestler .

SOMMAIRE

	PAGE
INTRODUCTION	IV.5
1 - Conception d'un projet scientifique et technique	IV.9
2 - Exemple de problèmes à résoudre	13
2.1. - Enoncé des problèmes	13
2.2. - Position du problème et buts à atteindre	14
2.3. - Comment atteindre le premier but fixé ?	15
2.4. - Pour atteindre le deuxième but	17
2.4.1 - Les formules	18
2.4.2 - Déroutement des calculs et signification des variables utilisées	18
2.4.3 - Les données du problème	20
2.5. - Comment est résolu le problème actuellement	22
2.5.1.- Premier aspect	22
2.5.2 - Deuxième aspect	22
3 - SOMINE permet-il de résoudre ces problèmes ?	25
3.1. - Le problème justifie-t-il l'implémentation d'une base de données ?	25
3.2. - Réponses apportées	26
4 - Solution utilisant une base de données SOMINE	29
4.1. - La structure Somine	29

4.2. - Le problème posé par la recherche du nom du matériau	31
4.3. - Des abaques aux programmes	31
4.3.1. - Une solution au problème des abaques	32
4.3.2. - Une solution plus générale	33
4.3.3. - La caractéristique programme	34
4.4. - La caractéristique programme et les actions implicites	36
4.5. - La caractéristique valeur+programme	39
4.6. - Généralisation	41
5 - Réalisation pratique de l'application	43
5.1. - Mise en place des caractéristiques programme et valeur +programme au niveau de la structure des informations	43
5.2. - Mise en place des caractéristiques programme et valeur + programme au niveau des requêtes	44
5.3. - La mise en place de la structure	45
5.4. - Les données à mettre dans la base SOMINE	48
5.5. - Les données des programmes d'utilisation	48
5.6. - Le programme d'aiguillage	51
6 - Les programmes ayant accès à la base Somine créée	53
6.1. - Schéma général	53
6.2. - Programme de mise à jour de la sous-base expérimentale. Programmes statistiques	55
6.3. - Calcul du coefficient de sécurité en mode conversationnel	56
6.4. - Calcul de α en traitement par lots de programmes	57

6.5. - Un exemple d'action implicite:	57
7 - Résultats et conclusion	59
8 - Bibliographie	65

INTRODUCTION

La conception assistée par ordinateur des projets scientifiques et techniques (C.A.O.) est née de l'envie d'appliquer la nouvelle science informatique à un autre domaine des activités humaines, elle provient également de l'intuition (vite vérifiée par l'expérience et les essais) de tout ce que pourrait apporter cette nouvelle science dans ce domaine.

L'implémentation du système de base de données SOMINE constituait une réflexion sur les gestions de mémoire possibles, les structurations et les accès aux informations dans une base de données. Cette partie de la thèse est une application de cette réflexion au domaine réel. Comme le dit J. ARSAC /12/ " Sans informatique appliquée, la recherche fondamentale risque de se stériliser et de s'enliser dans des discussions bizantines . Les problèmes viennent du front des applications".

Si dans les autres parties de cette thèse nous avons utilisé une "méthode de conception descendante " WIRTH /11/ , au contraire cette partie va mettre en évidence " une méthode de conception informatique ascendante " DERMIANE /10/, c'est-à-dire dans laquelle il va falloir améliorer, modifier la version de base de SOMINE pour arriver à une solution du problème réel que nous avons souhaité résoudre.

Diverses réalisations ont été faites ou sont en cours de recherche ou d'amélioration dans le domaine de la C.A.O. Nous citerons d'abord le vaste projet CIVA /10/ " qui vise à donner à l'utilisateur la possibilité d'employer une formulation unique pour représenter les différentes phases de la mise en oeuvre d'un travail ". On peut espérer en effet qu'une meilleure compréhension entre les nombreux services

mis à contribution dans la conception et la réalisation d'un traitement d'information permettrait de mieux situer et diviser le problème à résoudre. Toute décomposition d'une difficulté en sous-problèmes plus simples est toujours un pas de fait dans sa résolution.

J.C. LATOMBE /9/ a élaboré un système informatique pour l'assistance à l'enseignement et à la conception en électronique. Nous avons lu avec intérêt les publications faites à ce sujet car l'analyse des différents types de problèmes qui se posent dans la conception et les différentes réalisations possibles ont servi notre réflexion. Dans ce projet " il s'agit d'apprendre à des élèves ingénieurs les principes de la conception des machines électriques dont la constitution est très complexe". On peut voir apparaître à l'occasion de cette citation bibliographique la relation qui existe entre C.A.O. et E.A.O., il s'agit dans les deux cas d'appréhender un certain nombre de relations et de décomposer les difficultés afin de les mieux cerner /18/.

Nous pouvons aussi citer l'emploi des techniques de C.A.O. par BRACCHI et SOMALVICO /13/ dans le domaine des circuits électroniques.

Puis une visite à l'équipe de recherche de M. FEUVRAIS (Arts et Métiers - PARIS) nous a permis de prendre connaissance de la vaste synthèse de l'utilisation de la C.A.O. qu'ils sont en train de réaliser. Le but de cette équipe est de rassembler un certain nombre de réalisations existantes, d'écrire les logiciels et les interfaces entre ces réalisations et une base de données. Ils ont aussi mis en évidence pour nous l'intérêt du bureau d'étude et de la phase de dessin ou de modélisation dans la C.A.O. Un de leurs buts est de faire de la conception de forme et d'y associer des programmes de calculs en vue d'aider le concepteur.

Enfin c'est la connaissance de la création de l'association MICADO (Mission pour la Conception Assistée et le Dessin par Ordinateur) qui nous a montré l'intérêt grandissant de nombreux chercheurs pour la C.A.O. .

Leur regroupement montre leur volonté de réaliser des actions concertées et efficaces. Nous avons conscience de notre peu d'expérience en ce domaine mais nous espérons que les réflexions et les réalisations qui vont suivre nous permettront tout de même d'apporter une contribution dans ce vaste domaine en voie d'informatisation .

A la suite de cette étude bibliographique nous avons essayé de faire notre propre analyse de l'utilisation d'une base de données dans le domaine de la conception scientifique et technique.

1 - CONCEPTION D'UN PROJET SCIENTIFIQUE ET TECHNIQUE

Concevoir un projet scientifique et technique, c'est résoudre l'ensemble des problèmes qui se posent entre l'idée et la réalisation ou l'utilisation du projet devenu objet.

La conception commence par une phase de formalisation pendant laquelle est défini de façon précise le besoin à satisfaire par le matériel futur. Cette phase ne s'achève pas sans un tour d'horizon de tout ce qui se fait déjà dans ce domaine ou dans les domaines voisins.

Ensuite il y a une phase technologique pendant laquelle le projeteur choisit, conçoit et assemble les composants du projet.

Les problèmes à résoudre peuvent être classés en quatre grands types :

- les problèmes de recherche documentaire :

Entrent dans cette catégorie la recherche des solutions existantes au problème posé ou à des problèmes voisins ainsi que les contraintes et les hypothèses qui ont été retenues pour arriver à ces solutions. Dans ce type se place aussi la recherche des informations nécessaires à la conception du futur projet.

- les problèmes de résolution graphique :

Traçé de l'avant projet, des différentes solutions possibles et enfin de celle qui sera retenue /7/.

- les problèmes de résolution numérique :

Chaque résolution numérique se divise en plusieurs parties :

- * recherche d'une méthode de résolution ou algorithme,
- * description de cette méthode par un organigramme et enfin
- * résolution automatique.

Pour chacune de ces parties il peut y avoir recherche de solutions existantes ou création de programmes spécifiques.

- les problèmes relatifs aux essais et à la simulation du fonctionnement :

Ceux-ci sont destinés à vérifier la validité de la conception avant la réalisation industrielle.

Une importante classe de problèmes scientifiques nécessite la connaissance d'un certain nombre d'informations à extraire parmi un grand nombre de données. Souvent ces informations sont structurées de telle sorte que les bases de données de type hiénarchisées comme SOMINE paraissent particulièrement adaptée pour les mémoriser.

Outre cet aspect recherche documentaire, les problèmes de ce type font aussi appel à des techniques de résolution numériques et (ou) à des techniques de simulation dont la mise en oeuvre demande des moyens automatiques importants.

La base de données parait donc une structure particulièrement bien adaptée à ce type de problèmes. D'ailleurs des exemples d'utilisation existent déjà /22/, les services rendus sont de deux types :

* elle peut être un centre de documentation toujours disponible. En effet, elle peut contenir un assez grand nombre d'informations relatives à un sujet donné. Celles-ci sont agencées à l'intérieur de la base de façon à ce qu'il soit possible de les retrouver plus rapidement que par les moyens manuels de documentation. Plusieurs structurations peuvent être envisagées ce qui augmente les facilités de recherche des informations

qu'elle contient. Un autre avantage est que les informations stockées peuvent être de nature très différentes, ce qui permet de retrouver (ou de ranger) aussi bien le nom d'un matériau que la valeur d'un paramètre numérique, une liste de programmes ou de réalisations scientifiques déjà existantes, ou bien un élément standardisé de dessin.... Cette facilité est appréciable car elle permet d'avoir sur le même support toutes les informations nécessaires à un ou une classe de problèmes donnés.

* elle peut également servir à simuler plusieurs solutions à un même problème. C'est alors la rapidité de calcul de l'ordinateur qui permet de démarrer un processus itératif qui peut converger vers la solution optimale du problème posé. La base contient alors les données nécessaires aux différents essais et pouvant recueillir les résultats de ces essais. Est considérée comme optimale une solution qui vérifie certains critères codés ou chiffrés et qui ont pu être eux-mêmes introduits dans la base. C'est dans ce type d'utilisation que peuvent être placés les problèmes pour lesquels le projeteur envisage plusieurs solutions correspondant à des jeux de données différents..... la solution retenue sera celle qui donne les meilleurs résultats techniques ou économiques.

Généralement les aspects documentaires sont dissociés de ceux qui concernent les calculs ou les simulations.

SOMINE permet d'effectuer ces deux types de traitement simultanément. Grâce à des critères fournis par l'utilisateur ou contenus dans la base, il est possible, dans le cas de la simulation de choisir la meilleure solution au problème posé. Une des qualités de SOMINE est, en effet, d'être un langage complémentaire aux langages évolués (FORTRAN ou COBOL par exemple). Ceci signifie qu'à l'intérieur de n'importe quel programme écrit dans ces langages, on peut réaliser un ou plusieurs accès à la base ceci afin de placer une information ou bien d'en extraire une contenue dans la base ou créée à partir de celles qui y sont contenues. Les accès à la base peuvent être imposés par le programmeur ou dépendre des résultats de calculs intermédiaires ou de tests numériques ou logiques contenus dans le programme et qui sont les critères du problème.

2 - EXEMPLE DE PROBLEMES A RESOUDRE

Il faut que le problème que nous allons essayer de résoudre comporte de la recherche documentaire, des simulations à effectuer et des résolutions numériques (programmes de calcul). Le paragraphe suivant expose un problème de ce type et décrit les améliorations qu'il a fallu apporter à la version de base de SOMINE pour arriver à résoudre le problème compte - tenu des hypothèses et des critères retenus.

2. 1. - ENONCE DES PROBLEMES :

Sensibilisés par les problèmes scientifiques de mécanique et résistance des matériaux, nous avons choisi un problème dans cette dernière spécialité.

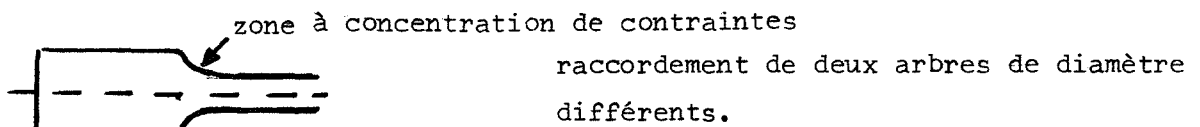
C'est un problème qui préoccupe plusieurs bureaux d'études, il s'agit de concevoir des ensembles de pièces métalliques (des structures) ayant une tenue en service convenable /16/ lorsqu'elles seront soumises à des sollicitations soit statiques soit dynamiques sinusoïdales, aléatoires ou impulsionnelles. Dans l'état actuel des travaux, des moyens informatiques (méthode des éléments finis par exemple) permettent d'évaluer le champ de contraintes dans une pièce soumise à des sollicitations extérieures. Ensuite, il y a détermination des critères de tenue et en particulier de la valeur du coefficient de sécurité. Actuellement cette détermination n'est pas optimisée et elle est souvent faite de façon empirique.

C'est dans le but de faire avancer d'un pas la détermination du coefficient de sécurité d'un ensemble de pièces métalliques que nous

.../...

avons pensé chercher une solution à ce problème.

Les laboratoires de recherche mécanique font des travaux permettant une meilleure détermination du coefficient de sécurité. Le C.E.T.I.M. (Centre Technique des Industries Mécaniques) a abordé le problème sous un angle assez simple en constatant que 50 % des ruptures en service sur les ensembles mécaniques avaient lieu sur des arbres soumis à des contraintes assimilables à des contraintes sinusoïdales de flexion, traction, torsion dans des zones à concentration de contraintes - en mécanique c'est une zone où la contrainte effective est supérieure à la contrainte nominale - par exemple :



Même dans ces cas simples, les bureaux d'études n'avaient pas de formules permettant d'évaluer le coefficient de sécurité. En particulier des abaques assez contradictoires suivant les auteurs ne permettaient pas de connaître l'influence de la forme, de l'état de surface

Grâce aux études bibliographiques ou expérimentales en cours au C.E.T.I.M. (études faites à partir des travaux de M. BAHUAUD, Directeur du Génie Mécanique de l'I.N.S.A.)/3/, nous sommes en mesure d'effectuer le calcul du coefficient de sécurité des pièces traitées dans la masse et soumises à des sollicitations sinusoïdales combinées de traction, flexion et torsion.

2. 2. - POSITION DU PROBLEME ET BUTS A ATTEINDRE :

Il s'agit d'étudier les caractéristiques mécaniques des aciers de construction dans le but de rechercher la valeur du coefficient de sécurité, problème qui présente deux aspects :

1 - les impératifs d'allègement des structures et de fiabilité accrue imposent aux bureaux d'études la connaissance de données statistiques sur la tenue des matériaux traités et sur l'influence (exprimée en termes statistiques) sur ces tenues des paramètres : forme, dimension, surface.

.../...

Or, à l'heure actuelle, le projeteur, même dans le cas de sollicitations aussi simples que la traction statique ou la traction dynamique, ne dispose que de valeurs moyennes et s'il a quelques renseignements sur les facteurs d'influence, ils sont en général flous et contradictoires. Il est donc obligé de choisir des valeurs qui le mettent en sécurité et qui supposent dépenses non justifiées en matière et main d'oeuvre.

Pourtant l'étude de la bibliographie montre que les résultats expérimentaux existent mais malheureusement sous forme fragmentaire qui nécessite une synthèse que l'on pourrait dès à présent baser sur l'étude d'une cinquantaine d'ouvrages et d'un millier d'articles.

2 - Une fois ces données statistiques obtenues, le projeteur devrait pouvoir y accéder facilement, soit de façon directe, soit sous forme d'utilisation de programmes de calcul ayant eux-mêmes directement accès à ces informations. L'interrogation de l'ensemble programmes, données s'effectuant suivant un schéma conversationnel permettant une optimisation aisée des calculs.

Nous avons donc deux buts à atteindre :

* à l'aide de moyens statistiques avoir une meilleure connaissance :

- de l'influence de certains paramètres chimiques, mécaniques ou métallurgiques sur la tenue statique et dynamique des matériaux.
- de ces paramètres par une synthèse critique des résultats existants ou obtenus.
- des correlations qui existent entre les caractéristiques mécaniques

* Appliquer ces connaissances au calcul des structures mécaniques .

2. 3. - COMMENT ATTEINDRE LE PREMIER BUT FIXE ?

Il faut tout d'abord être en possession des données expérimentales nécessaires aux calculs statistiques. Ces données sont issues :

.../...

- soit des documents imprimés existants qui constituent la somme des essais déjà faits,

- soit d'essais à faire sur des éprouvettes ou des structures métalliques existantes.

Dans le cadre de ce projet, nous avons retenu un certain nombre de données expérimentales. Elles nous ont semblé les plus représentatives de l'application que nous voulons mettre en place, naturellement leur nombre pourrait être augmenté si la nécessité s'en faisait sentir. Les mesures logiques faites sur les bases de type SOMINE permettent de penser qu'une telle augmentation ne dégraderait pas les performances de la base. Mais cette remarque montre que si une base doit être utilisée dans le domaine scientifique une de ces qualités doit être la possibilité d'effectuer des rajouts de structure car les problèmes de ce type sont toujours en évolution. Dans l'état actuel du prototype la structure est statique mais une greffe de structure pourrait être **faite** par exemple en prévoyant un nœud de l'arbre structure d'un type différent et qui contiendrait les pointeurs vers la sous structure rajoutée. Les travaux se poursuivent dans ce sens.

Voici la liste des caractéristiques qui pourront être fournies à l'occasion de chaque essai déjà réalisé ou à effectuer

Identification du matériau

* le nom du matériau considéré, sa composition chimique et le traitement thermique qu'il a subi.

* Le numéro, la date et le lieu de l'essai considéré .

* la caractéristique de trempabilité JOMINY : duretés de Rockwel -C (J10, J15, J 20)

Essais statiques

- Traction

* Le numéro, la date et le lieu de l'essai considéré .

* la résistance à la rupture : R

* la limite élastique apparente : R_e

* A % l'allongement à la rupture

* R_R : la résistance vraie à la rupture

- Resilience

* Le numéro, la date et le lieu de l'essai considéré .

* Les résiliences sur éprouvettes entaillées en U et en V à 20 °

Essais dynamiques

- * Le numéro, la date et le lieu de l'essai considéré .
- * Le type d'essai dynamique .
- * La forme de l'éprouvette.
- * Le mode de sollicitation et d'usinage de l'éprouvette.
- * La rugosité totale R_t .
- * Le coefficient de concentration de contrainte K_t
- * σ_M : la contrainte statique moyenne .
- * La limite de fatigue σ_b ou σ_{-1} et l'écart type de la limite de fatigue.
- * le mode de détermination de la limite de fatigue .

A partir de ces données devront être réalisés :

* des traitements statistiques en vue de l'obtention pour un matériau et un traitement thermique donné de valeurs moyennes et probabilistes.

* l'établissement de corrélations permettant de mettre en évidence les différents facteurs d'influence : état de surface, forme, dimension.

* une comparaison critique des résultats précédents avec les données bibliographiques existantes.

2. 4. - POUR ATTEINDRE LE DEUXIEME BUT :

C'est à dire l'optimisation du doublet (tenue en service, prix de revient) d'une structure donnée ; pour cela on doit entreprendre des calculs scientifiques utilisant :

- les valeurs moyennes et probabilistes obtenues au paragraphe précédent.
- les corrélations dont nous avons déjà parlé
- les données propres à la structure étudiée (forme, champ de contraintes)

Les formules et les conventions que nous avons adoptées dans le cadre de cette première approche sont définies dans les paragraphes suivants .

.../...

2. 4. 1. - Les formules :

Nous avons utilisé les formules suivantes issues des références bibliographiques /3/ et /4/, tout en introduisant (pour une première approche) un certain nombre de simplifications .

$$(1) \quad \alpha = R_e / \sigma_{eq. tot.}$$

$$(2) \quad \sigma_{eq. tot.} = \sqrt{\sigma_{eq.}^2 + 3\tau_{eq.}^2}$$

$$(3) \quad \sigma_{eq.} = K_f \sigma_{stat.} + \frac{K_t}{\epsilon_s \cdot \epsilon_{ech.}} \sigma_{dyn.} \frac{R_e}{\sigma-1}$$

$$(4) \quad K_f = 1 + q (K_t - 1)$$

$$(5) \quad \tau_{eq.} = K'_f \tau_{stat.} + \frac{K'_t}{\epsilon'_s \cdot \epsilon'_{ech.}} \tau_{dyn.} \frac{R_e}{\sigma \cdot D}$$

$$(6) \quad K'_f = 1 + q (K'_t - 1)$$

2. 4. 2. - Déroulement du calcul et signification des variables utilisées :

- C'est la formule (1) qui donne la valeur du coefficient de sécurité α cherché.
- R_e représente la limite élastique du matériau considéré
- $\sigma_{eq. tot.}$ représente la contrainte totale équivalente à un ensemble de contraintes nominales et tangentiels en un point considéré de la pièce. Sa valeur est déterminée par le critère de Von Mises (formule 2) dans lequel apparaissent deux variables nouvelles σ_{eq} et τ_{eq} .
- σ_{eq} est la contrainte normale statique dont l'action est équivalente à l'action des contraintes normales statiques et dynamiques.
- τ_{eq} est la contrainte tangentielle statique dont l'action est équivalente à l'action des contraintes tangentielles statiques et dynamiques.

.../...

Ce sont ces formules (3) et (5) qui permettent de trouver les valeurs de σ_{eq} et τ_{eq} . Dans ces formules nous trouvons les variables suivantes :

- K_t coefficient de concentration de contraintes théoriques en fond d'entaille en sollicitations normales
- K'_t coefficient de concentration de contraintes théoriques en fond d'entaille en sollicitations tangentiellles
- ξ_s coefficient d'état de surface en sollicitations normales
- ξ'_s coefficient d'état de surface en sollicitations tangentiellles
- ξ_{ech} coefficient d'échelle en sollicitations normales.
- ξ'_{ech} coefficient d'échelle en sollicitations tangentiellles
- σ^{-1} limite de fatigue du matériau considéré. (ou σ_D)
- σ_{stat} contrainte normale nominale moyenne
- τ_{stat} contrainte tangentielle nominale moyenne
- σ_{dyn} contrainte normale nominale alternative
- τ_{dyn} contrainte tangentielle nominale alternative

Ce sont les formule(4) et (6) qui permettent de calculer les valeurs de K_F et K'_F connaissant K_t et K'_t et la valeur de la variable q . q représente la sensibilité à l'entaille du matériau et c'est l'indice de sensibilité d'entaille en sollicitations normales et tangentiellles.

K_F et K'_F sont les facteurs effectifs de concentration de contraintes .

Les formules du paragraphe 2.4.1. s'appliquent à la théorie des poutres dans le cas de matériaux homogènes pour lesquels la limite élastique en compression égale la limite élastique en traction.

2. 4. 3. - Les données du problème à résoudre :

On peut grouper les variables énumérées au paragraphe précédent en trois catégories :

- celles qui sont données par le concepteur et qui sont liées à la pièce considérée.
- celles qui sont également liées à la pièce mais sont le résultat de calculs antérieurs.
- enfin celles qui sont issues des traitements définis en 2.3.

Nous allons classer les données du problème à résoudre dans les trois catégories précédemment citées, nous verrons ainsi la nature des données qui devront faire l'objet d'une recherche dans la base de type SOMINE que nous voulons mettre en place.

* Les données σ_{stat} , τ_{stat} , σ_{dyn} , τ_{dyn} , sont calculées dans le bureau d'études où est conçue la pièce. Dans les cas simples ce calcul précède immédiatement celui du coefficient de sécurité. Dans les cas plus complexes le calcul est fait automatiquement en utilisant par exemple un algorithme de calcul par éléments finis. La difficulté de ce calcul dépend de la forme de la pièce et de la nature des sollicitations. Quelque soit le cas, nous pouvons réaliser un lien entre le calcul et celui du coefficient. Dans la suite de ce problème, nous considérerons que les variables σ_{stat} , τ_{stat} , σ_{dyn} , τ_{dyn} , sont des données résultant d'un calcul antérieur.

* Les autres variables dont nous avons donné la signification dans le paragraphe précédent sont le résultat des traitements statistiques dont il a été question au paragraphe 2.3

* Les données nécessaires à la recherche de la valeur des variables entrant dans la composition des formules du paragraphe 2.4.1. vont être énumérées maintenant. Il s'agit :

- du nom du matériaux utilisé et du traitement thermique qu'il a subi, ces valeurs permettent de déterminer R_e (limite élastique) et σ_D (limite de fatigue)

- du mode de sollicitation auquel est soumise la pièce permettant de déterminer par recherche dans des abaques les valeurs de K_t , ϵ_{ech} , K'_t , ϵ'_{ech} .

- Pour la détermination de K_t et K'_t il est nécessaire de connaître la forme de l'entaille au point considéré.

par exemple pour la forme ci-dessous, celle-ci est caractérisée par :

- t profondeur de l'entaille
- a rayon de la pièce au fond de l'entaille
- r rayon de l'entaille

le schéma ci-dessous est un exemple des caractéristiques t, a, r.

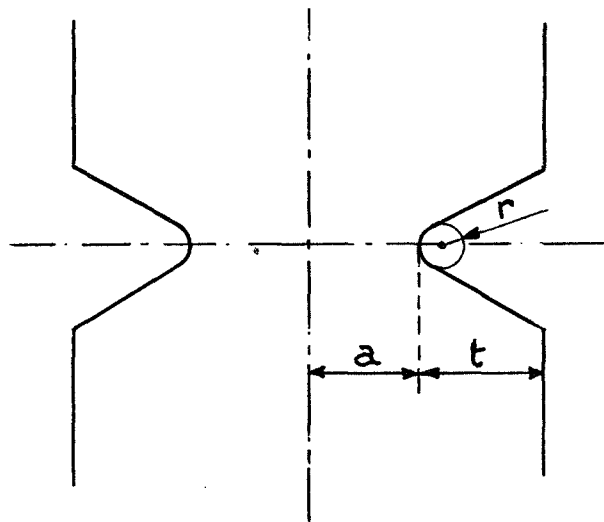


Figure IV.1

Le rapport a/t définissant - les entailles profondes si $a/t \rightarrow 0$
 - les entailles plates si $t/a \rightarrow 0$

La donnée du mode d'usinage ainsi que celle de la rugosité de la pièce permettent de trouver ϵ_s et ϵ'_s .

2. 5. - COMMENT EST RESOLU LE PROBLEME ACTUELLEMENT :

2. 5. 1. - Premier aspect.

Dans l'état actuel des choses les résultats des essais figurent dans des notes isolées ou dans une abondante documentation imparfaitement dépouillée et rassemblée.

La comparaison des résultats, de leur vraisemblance et de leur compatibilité n'est pas faite. Enfin aucune étude statistique ou critique de toutes les informations existantes n'a été entreprise.

En résumé, le premier but que nous nous sommes fixés n'a jamais été atteint. Ceci parce que la nécessité ne s'en était encore jamais fait sentir, seules des synthèses partielles ont été réalisées et, c'est la mise en oeuvre de moyens informatiques qui permet d'envisager une solution globale.

Les différentes étapes de la méthode qui permettrait de résoudre ce problème ont été décrites au paragraphe 2.3. on peut les résumer ainsi :

- collecte des résultats relatifs à l'essai en cours,
- traitements statistiques,
- établissement de corrélations,
- critique des résultats obtenus,
- fin de l'expérience ou nouvel essai.

2. 5. 2. - Deuxième aspect :

Quant au deuxième but dans le cas où les pièces sont calculées, on applique un coefficient obtenu à priori ou souvent défini par l'expérience.

.../...

Les recherches effectuées dans ce domaine permettent dans les hypothèses que nous avons considérées d'appréhender le problème en utilisant les formules du paragraphe 2.4.1. Elles contiennent les données déjà énumérées et dont nous avons expliqué les relations et les modes de recherche dans le paragraphe précédent.

En utilisant ces formules par exemple le calcul du coefficient de sécurité α se fait manuellement et il nécessite des recherches d'abaques ou de valeurs dans des tables ou des listes.

Il semble que cette méthode qui demande du personnel et un certain temps pourrait être optimisée par une résolution automatique. Pour cela nous allons essayer de décrire un algorithme de résolution de ce problème et de le représenter ensuite par un organigramme.

Méthode d'optimisation de la valeur de α

La méthode que nous devons décrire comportera de la recherche documentaire et des calculs numériques, pour arriver à une valeur acceptable du coefficient de sécurité il conviendra de se donner une plage de valeurs pour α ainsi que des impératifs économiques ou des critères de construction.

On peut envisager le calcul de plusieurs valeurs de α obtenues avec les valeurs extrémales ou moyenne des paramètres, chacune d'elles étant accompagnée de sa probabilité.

Les itérations se feront sur les valeurs de α et le prix de revient du projet, on envisagera d'abord des modifications dans le choix du matériau ainsi que dans le traitement thermique et en dernier lieu des modifications dans les dimensions de la pièce conçue.

Le schéma de la page suivante constitue une description de cette méthode

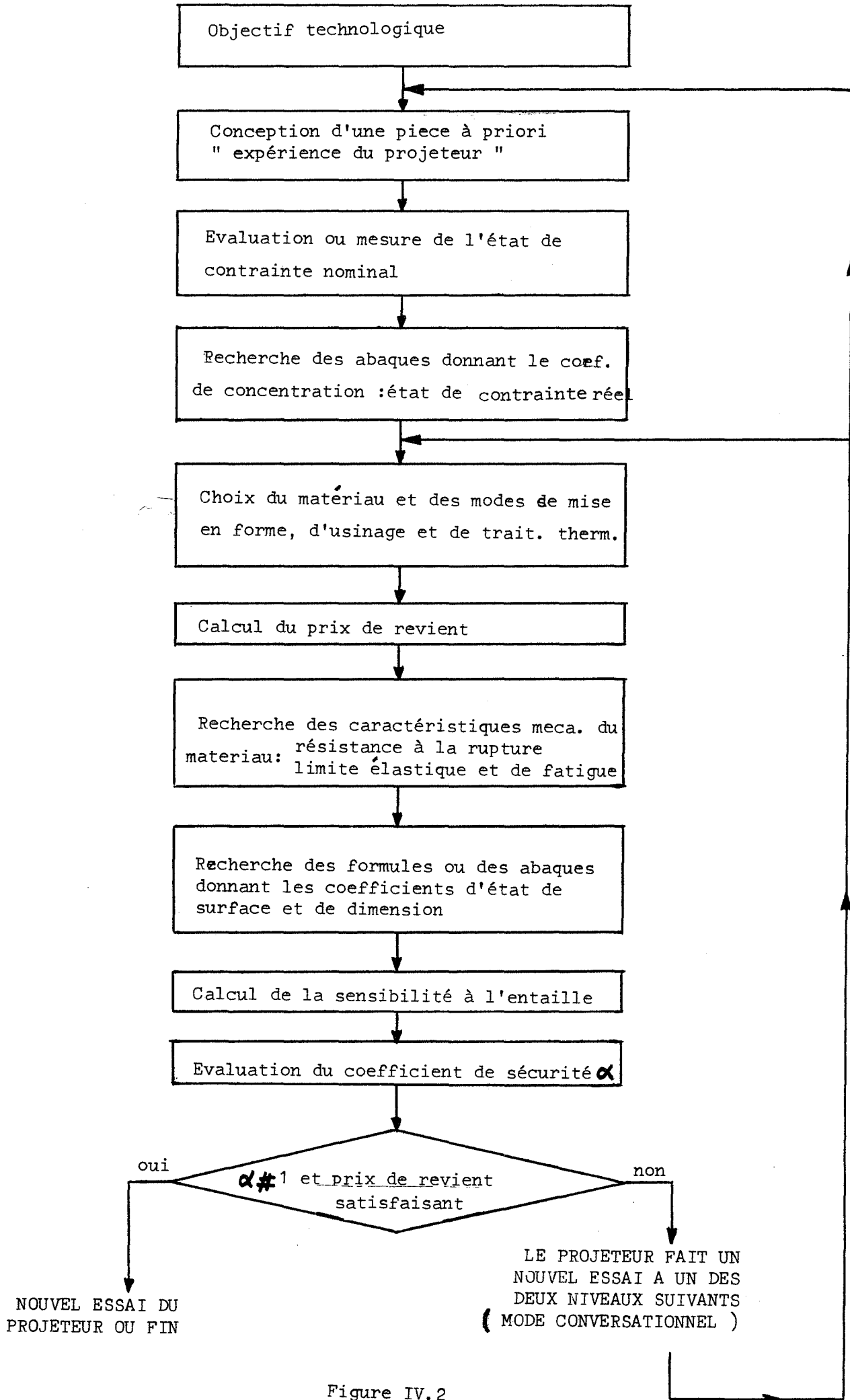


Figure IV.2

3 - SOMINE PERMET-IL DE RESOUDRE CES PROBLEMES ?

Pour avoir une réponse affirmative à cette question, il faut que :

- d'une part le problème justifie l'implémentation d'une base de type SOMINE
- d'autre part les problèmes posés par sa résolution soient tous solubles à l'aide de la base.

3. 1. - Le problème justifie -t-il l'implémentation de la base ?

* Pour cela il faut que le nombre d'informations ou de données qui vont être manipulées soient en nombre suffisant. En effet, pour nécessiter une structuration telle que celle de SOMINE, il faut envisager un problème pour lequel le temps d'accès à une information située dans un fichier séquentiel serait plus grand que celui permettant de la trouver dans le fichier de SOMINE.

* Non seulement le nombre d'informations est important mais également leur nature, car n'oublions pas que dans une base SOMINE des textes, des valeurs numériques entières ou réelles, des graphiques..... peuvent voisiner.

* Les différentes caractéristiques d'une base SOMINE sont structurées entre elles et représentées par un arbre que la présence de REFERENCES transforme en graphe orienté. Il faut que cette structuration soit

possible et qu'une structure moins riche : tableau ou liste ... ne permette pas de résoudre les problèmes d'une façon satisfaisante du point de vue de la place mémoire ou de l'accès aux informations.

Les questions à se poser sont donc les suivantes :

pourra t-on organiser hiérarchiquement les différentes informations nécessaires à la résolution du problème ?

ou bien est-ce qu'une structure arborescente traduira l'ensemble des relations qui existent entre les différentes données ?

En effet, si le problème comportait plusieurs séries de données indépendantes les unes des autres dans ce cas un fichier séquentiel indexé résoudrait le problème de la recherche.

Inversement si l'ensemble de données A est en relation avec l'ensemble de données B et que cette relation est symétrique, le problème ne peut être résolu que dans un certain nombre de cas à l'aide de la caractéristique INVERSE.

* Enfin, l'implémentation d'une base serait justifiée si les modes de mise à jour et d'interrogation sont suffisamment souples et simples pour pouvoir être utilisés après un court apprentissage. Il faut également se préoccuper de l'ensemble des questions qui devront être posées à la base et si elles sont compatibles avec le système et utilisent ses possibilités.

3. 2. - Réponse apportée par SOMINE à ces quatre questions

* Pour justifier l'implémentation d'une base le premier point concernait le nombre d'informations. La résolution d'un problème scientifique ne brasse jamais un aussi grand nombre de données qu'un problème de gestion par exemple. Cependant, il semble que le nombre d'informations numériques ou alphanumériques à manipuler soit en nombre suffisant. En effet, on peut évaluer à une vingtaine de caractéristiques de ces types pour chaque essai dont on veut conserver les résultats. Enfin, plus de 2000 valeurs numériques réelles sont nécessaires pour retenir les valeurs moyennes et extrêmes des paramètres nécessaires au calcul de α et ceci dans le cas où

.../...

on se limite à 50 matériaux pouvant subir 5 traitements thermiques différents :

En effet si l'on retient :

2 valeurs extrêmes pour σ^{-1} et une valeur moyenne		X 50 X 5 = 2250
2 valeurs extrêmes pour R_e et une valeur moyenne		
2 valeurs extrêmes pour R et une valeur moyenne		
		valeurs différentes

* nous venons de parler des données numériques, pour faire un recensement total des informations nécessaires il importe de se rappeler que la solution du problème nécessite la réalisation et la consultation de nombreux abaques. Nous verrons au paragraphe 3.1 quelles solutions nous avons envisagées et celle que nous avons retenue pour permettre cette réalisation et cette consultation.

Naturellement, le problème pourrait être encore généralisé en augmentant le nombre de possibilité mais il nous est apparu suffisant de le résoudre dans ce cadre pour démontrer les possibilités de SOMINE. Sa généralisation pourrait se faire dans le cadre d'un ajout de structure lorsque cette possibilité sera offerte aux utilisateurs.

* Au niveau des relations qui existent scientifiquement entre les différentes caractéristiques de la future base nous avons fait les constatations simples suivantes :

- les informations ne sont pas indépendantes
- les relations qui les lient sont en petit nombre
- elles sont toujours non symétriques (si $A \Rightarrow B$, le contraire n'a pas lieu)

On trouvera des développements sur les relations qui peuvent exister entre les éléments d'une base dans /17/. Nous pouvons constater ici que ce n'est pas dans le domaine des applications scientifiques que ce problème présente le plus de difficultés.

* La quatrième question soulevée au sujet de la justification de la mise en place d'une base est celui de la mise à jour et de l'interrogation. Ces deux requêtes sont simples à mettre en oeuvre (voir partie III).

Comme il a été vu dans ce chapitre les accès à la base pourront avoir lieu soit à l'intérieur d'un programme de calcul écrit en FORTRAN soit à partir de l'écran qui fera office de collecteur de données et de support des résultats.

4 - SOLUTION UTILISANT UNE BASE DE DONNEES SOMINE

Les réponses aux quatre questions posées étant affirmatives nous allons essayer de construire une base SOMINE et les programmes qui permettront de résoudre le problème posé.

4. 1. - La structure SOMINE :

Pour arriver à construire la structure de la base recherchée nous nous sommes d'abord posé quelques questions de base où requêtes qui nous permettront d'avoir accès à la base.

voici quelques exemples :

créer l'essai 10

mettre à jour date de l'essai 10

interroger K_t de l'essai 10

interroger R moy du traitement thermique 2 du matériau 3

interroger K_t du mode de sollicitation 3

voici leur traduction en requête SOMINE :

C ESSAI 10 #

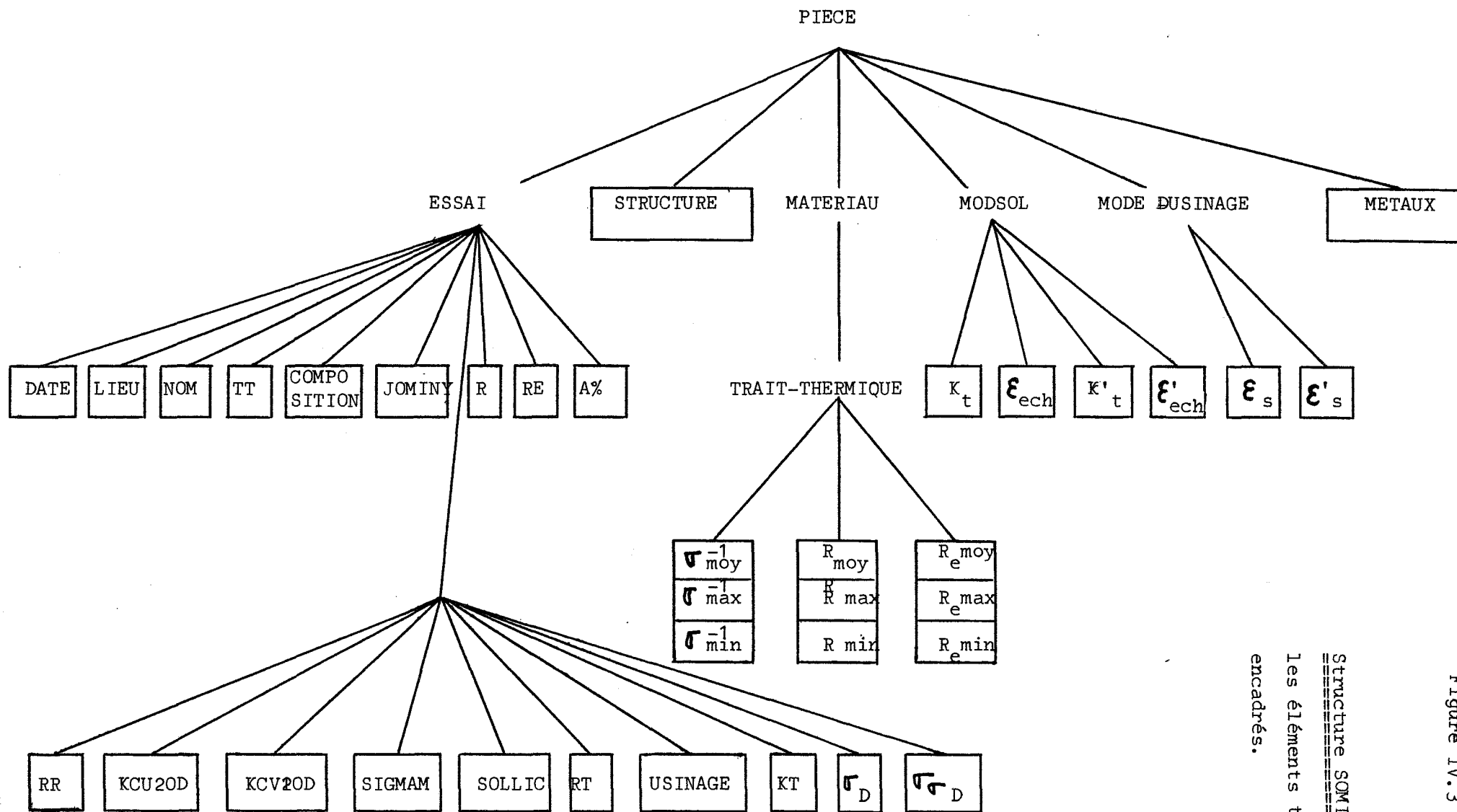
M DATE DE L'ESSAI 10 #

I K_t DE L'ESSAI 10 #

I RMOY DU TRAIT-THERMIQUE 2 DU MATERIAU 3 #

I K_t DU MODSOL 3 #

.../...



Structure SOMINE :
 les éléments terminaux sont
 encadrés.

Figure IV.3

En quelques essais nous avons réussi à construire la structure arborescente de la page précédente qui rend compte des relations existantes.

4. 2. - Le problème posé par la recherche du nom du matériau :

La recherche du nom du matériau nous a posé quelques problèmes.

En effet, le nom du matériau n'est pas une feuille de l'arbre représenté à la page précédente il fallait donc associer un numéro de réalisation de l'entité MATERIAU à chaque désignation d'un matériau. Cette méthode nécessite la connaissance par l'utilisateur de ce numéro ce qui va à l'encontre de notre souci de facilité d'utilisation. Nous avons donc placé la liste des matériaux dans la structure dans une caractéristique terminale de type texte désignée par METAUX : Ensuite, il n'y a plus qu'à procéder par comparaison avec les éléments de ce texte pour déterminer le numéro de la réalisation considérée, l'utilisateur n'a donc qu'à donner le nom du matériau (voir liste paragraphe 5.5)

Dans la plupart des cas, seuls les 4 premiers caractères suffisent à la détermination, on a à effectuer une recherche séquentielle sur les mots-mémoire qui contiennent la liste des matériaux stockée sous forme de texte SOMINE.

4. 3.- Des abaques aux programmes :

Le stockage des informations numériques ou alphanumériques dans une base SOMINE ne pose plus de problème /17/

Plusieurs solutions sont possibles pour le stockage des abaques.

Une des solutions au problème des abaques peut être la suivante : Ils peuvent être introduits dans la banque sous forme de graphes de fonctions à valeurs discrètes. Une caractéristique de type texte permet en effet de stocker les graphiques. C'est la première solution qui s'est présentée à l'esprit et c'est également la plus simple à mettre en oeuvre. Mais le problème réside dans l'interrogation d'une telle courbe.

En effet, il faudrait retrouver parmi une suite de points celui dont l'abscisse est la plus voisine de celle que l'on cherche ! Cette recherche peut être longue si l'on sait que pour obtenir une précision suffisante il faut considérer un grand nombre de points.

Une autre solution dérivée de la première consisterait à ranger les couples (abscisses, ordonnées) dans un texte et à faire une recherche du même type que la précédente. Les inconvénients de cette méthode sont les mêmes et de plus on ne peut pas avoir une vision globale de l'abaque sans écrire un programme supplémentaire.

4. 3. 1. - Une solution au problème d'abaques :

Une troisième solution consisterait à mettre les abscisses des points comme les numéros de réalisation d'une entité et les ordonnées comme valeurs d'une caractéristique fille de l'entité et contenue dans la base.

exemple :

supposons que les couples suivants sont les coordonnées de points d'un abaque.

(1, 3)
 (2, 3.2)
 (3, 3.5)
 .
 .
 .

écrivons le morceau de structure correspondant à la solution envisagée :

.
 .
 .
 ABAQUE (100)
 |
 ORDONNEE

L'interrogation d'une telle structure pourrait être la suivante :

I = 3
 I ORDONNEE DE L'ABAQUE X (I) DE

Le résultat de cette interrogation serait :

$$\text{ORDONNEE} = 3.5$$

Dans ce cas le problème des abscisses non entières peut être résolu en les multipliant par une puissance de 10 convenablement choisie.

Naturellement si on ne veut pas avoir à stocker les abscisses il faudra que dans la définition discrète de l'abaque le pas soit constant.

L'avantage de cette méthode est sa facilité à mettre en oeuvre, en effet il n'y a qu'à rajouter les caractéristiques ordonnées sous les caractéristiques abaques transformées auparavant en entités. Un inconvénient provient du fait que la valeur de la fonction ne peut être calculée qu'en des points précis. Ceux-ci peuvent naturellement être très nombreux pour augmenter la précision. On peut également calculer la valeur de la fonction aux deux points qui encadrent le point considéré et faire alors la moyenne des deux valeurs ou tout autre calcul permettant d'arriver à une meilleure précision. Cette méthode a été mise en oeuvre dans le cas du problème de résistance des matériaux que nous étudions, les résultats qu'elle a donnés sont indiqués au paragraphe 5.

4. 3. 2. - Une solution plus générale :

Le problème qui est posé à propos des abaques peut être envisagé d'une façon plus générale.

D'abord qu'est-ce qu'une équation de courbe ?

* C'est une ou plusieurs expressions arithmétiques (suivant que la fonction a une ou plusieurs déterminations suivant la valeur de la variable).

* C'est donc un cas particulier de programme.

Ce programme peut être écrit en langage évolué et situé dans la base ou bien appelé à partir de la base. /6/

Nous avons essayé à l'aide de SOMINE et à l'occasion de la résolution de ce problème de résistance des matériaux de trouver une solution au problème plus général suivant :

Comment faire exécuter un programme à partir d'un accès à la base, le programme dépendant des données contenues dans la requête ?

Plusieurs possibilités s'offraient à nous :

- * une solution consistait à ranger le programme dans la base en binaire. L'exécution pouvait être assurée à partir de cartes de contrôle générées à l'aide de macro-instructions. Cette solution nous a semblé lourde à mettre en oeuvre.

- * une autre possibilité dont la programmation aurait été assez longue est l'écriture d'un interpréteur de langage évolué (limité aux expressions arithmétiques dans un premier temps.) Il permettrait d'assurer l'exécution du programme au moment voulu.

- * une autre idée à laquelle nous n'avons pas donné de suite a été l'implantation des expressions arithmétiques dans la base à l'aide d'une arborescence de type SOMINE.

- * Enfin, l'idée que nous avons retenu et mise en oeuvre (avec celle du paragraphe 3.1. 1) est une solution simple et efficace dont le point de départ a été la création d'une nouvelle caractéristique simple que nous avons appelée Programme et que nous allons développer dans le paragraphe suivant.

4.3.3. - La caractéristique programme :

Les caractéristiques programmes sont des éléments terminaux au même titre que les textes, les valeurs numériques....

Ce nouveau type sera lié par les pointeurs père, frère; fils comme les autres caractéristiques simples.

Le code qui lui a été attribué est 15.

Dans le quinzième mot de la sous-page a été placé au moment de la création du fichier structure un numéro qui sera celui du ou des programmes situés au noeud de cet arbre.

Structurellement, une telle caractéristique est définie par le vecteur suivant :

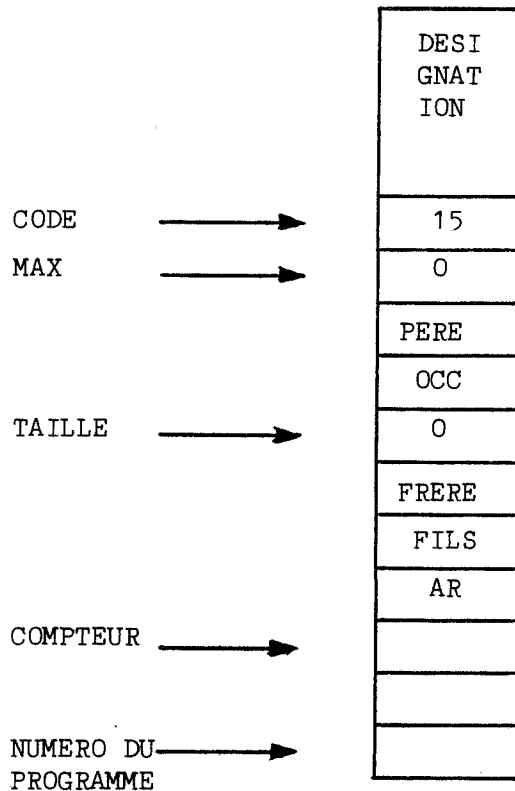


Figure IV.4

Soit la partie de structure ci-dessous

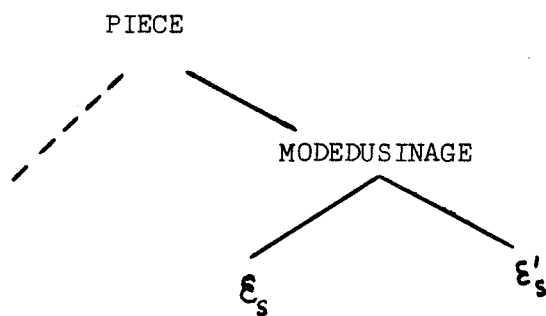


Figure IV.5

Si ε_s doit être calculé à l'aide d'un programme, comme nous avons retenu 5 mode d'usinage, 5 programmes permettront ce calcul. Pour faire le choix parmi les programmes possibles on utilise le tableau NUCIT (partie III paragraphe 5.1) qui contient les numéros de réalisation des entités

.../...

rencontrées sur le parcours de la requête.

On écrira donc un programme d'aiguillage utilisant le numéro structurel de la caractéristique "programme" et les valeurs de NUCIT, ce programme déclenchera un appel au programme à exécuter.

Les programmes écrits en langage évolué ou en autocode auront donc été indépendamment mis sur une bibliothèque en binaire. Ils n'occupent aucune place dans le fichier-données c'est pourquoi la taille des caractéristiques programme a été déclarée nulle.

* La mise à jour de la caractéristique programme se fait lors de la création de la structure par attribution d'un numéro d'ordre.

* L'interrogation d'une caractéristique programme se fait suivant le même modèle que les requêtes élémentaires définies à la page.

exemple :

I EPS DU MODEDUSINAGE 2

Cette interrogation appelle et exécute le sous-programme qui permet de calculer EPS pour le mode d'usinage numéro 2. Le résultat sera dans ce cas un numérique réel contenu dans la variable RNUME et sorti sur l'imprimante, sur l'écran ou formaté (suivant la valeur de IAPEL)

La caractéristique programme et son utilisation permettent de donner une réponse à la question :

Comment faire exécuter un programme à partir d'un accès à la base ?

Voyons maintenant quels sont les défauts, les possibilités et les qualités de cette caractéristique.

4. 4. - La caractéristique programme et les actions implicites :

- le langage SOCRATE ne permet pas toutes les opérations arithmétiques, de comparaison ou de relation ce qui rend impossible les calculs types calculs de bureau ou calculs scientifiques. Ceci est dû au fait que les

variables Socrate sont toujours de type entier et que les opérateurs arithmétiques en particulier ne font pas partie du langage de requêtes.

- enfin le problème des actions spontanées (c'est à dire les modifications des informations contenues dans la base provoquées en chaîne par une seule mise à jour) n'a pas trouvé de solution dans Socrate.

Depuis la partie II, nous savons que les calculs de toutes natures sont possibles pour SOMINE puisqu'on peut trouver dans une base de cetttype des caractéristiques numériques réelles ou entières, positives ou négatives. De plus tous les calculs ou comparaisons peuvent être effectués dans le langage évoqué dans lequel sont immergés les accès à la base.

Voyons comment la caractéristique PROGRAMME permet d'apporter une solution au second problème.

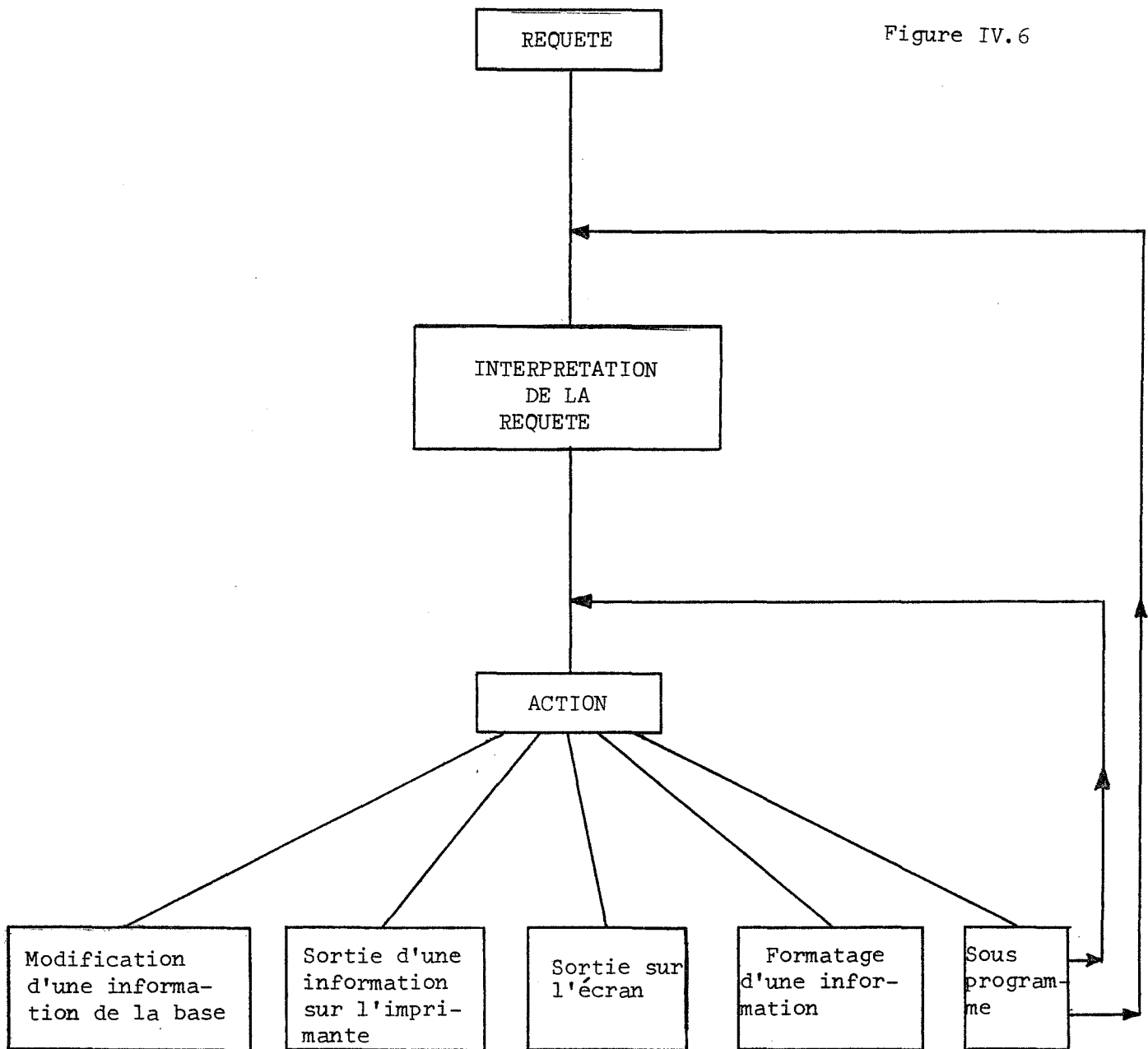
En effet, supposons que l'on a un programme utilisateur contenant des requêtes portant sur une caractéristique de type programme. Si la caractéristique programme fait exécuter le programme SP1 et dans le cas où SP1 est écrit en Fortran, il peut contenir : des instructions arithmétiques, des appels à des sous-programme extérieurs à la base et aussi des instructions CALL INTER (requête). C'est - à - dire, qu'il peut y avoir modification d'une ou plusieurs caractéristiques de la base ceci spontanément à partir de l'exécution d'une requête. Il y a donc actions implicites avec tous les risques mais aussi tous les avantages que cette notion présente (nous en verrons un exemple au paragraphe 6.5) . Les risques d'une telle mise en place ne sont pas négligeables car une caractéristique programme faisant elle-même appel à la base peut y provoquer des modifications qui n'avaient pas été prévues en première analyse.

La caractéristique programme permet par son exécution de déclencher des actions en chaîne dans la base ou à l'extérieur. On a donc mis en place un phénomène récursif d'actions qui sont entreprises soit sur les éléments de la base soit sur les données du problème.

Les actions qui peuvent être entreprises sont toutes celles que peut effectuer un langage évolué c'est-à-dire évaluation d'expressions

arithmétiques, logiques, ou de relations, tests, boucles, appels à des procédures externes, entrées et sorties d'informations.

Le processus récursif de l'utilisation des requêtes peut se schématiser ainsi :



4. 5. - La caractéristique valeur + programme :

On peut envisager le cas où le résultat d'un programme lié à la base SOMINE est un élément d'information nécessaire non seulement à l'utilisation du moment mais que l'on aimerait également pouvoir conserver dans la base.

Ce problème pourrait se traiter si le besoin s'en faisait sentir dans toute sa généralité. Dans le cadre de l'aide à la conception des projets scientifiques et techniques les résultats sont le plus souvent des numériques réels. Nous nous bornerons donc au cas où le programme ne donne qu'un seul résultat de type numérique réel. Dans ce cas une petite modification de la caractéristique programme permet de résoudre ce problème.

Nous avons vu que pour le type programme la taille de l'emplacement réservé dans le fichier-données est nulle puisque le programme à exécuter se trouve sur une bibliothèque utilisateur. Pour créer la nouvelle caractéristique valeur + programme il suffit de réserver dans le fichier-données un emplacement qui pourra contenir le résultat du programme. Dans le cas présent la taille réservée sera un mot comme pour toutes les caractéristiques numériques réelles.

Cette dernière possibilité permet d'avoir connaissance de la valeur d'une caractéristique avant et après l'exécution du programme. On peut ainsi voir l'impact d'une modification sur une information de la base.

exemple :

Si un programme calcule le prix de revient d'une pièce composée de :

```

n1  constituants  c 1 au prix p1
n2  constituants  c 2 au prix p2
:
.
ni  constituants  c i au prix pi

```

Si le prix de revient de la pièce est une caractéristique de type valeur + programme, l'utilisateur pourra interroger l'ancien prix de revient et le prix de revient obtenu après des modifications sur les prix ou le nombre des constituants par exemple.

L'implantation dans la structure d'une telle caractéristique ne provoque qu'une modification au niveau de la taille occupée dans le fichier-données par l'information en question.

Outre l'attribution d'un numéro structurel au programme, la mise à jour se fera par une requête analogue à celle portant sur les informations numériques réelles.

Par contre, un nouveau mode d'interrogation doit être mis en place. En effet, comme il a été vu dans l'étude de la caractéristique programme (voir paragraphe 4.3.3.) le mode habituel ne provoque que l'exécution du programme. La lecture préalable de l'information et le dépôt du résultat du programme doivent donc être déclenchés par un ordre différent. Si nous regardons l'automate de définition de requêtes (Part.III) la simplicité de sa conception permet d'ajouter un nouveau mode sans entraîner de grandes modifications de programmation.

Il suffira de rajouter une transition supplémentaire à l'automate :

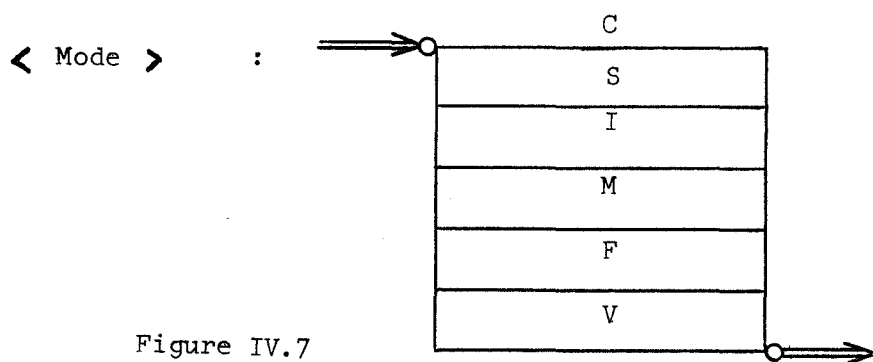


Figure IV.7

Le nouveau mode valeur déclenchera une lecture dans le fichier-données l'exécution du programme qui s'y rattache et le remplacement de l'information par le résultat du programme (une sortie de résultat peut éventuellement être prévue).

4. 6. - Généralisation :

On peut se demander si les caractéristiques programme et Valeur+programme ne sont pas des cas particuliers d'une caractéristique plus générale.

En fait, elles constituent deux approches différentes des améliorations qui peuvent être apportées à la base. Elles mettent en jeu les qualités de modularité de la programmation et l'évolutivité du produit SOMINE.

La caractéristique Programme est la création d'un type de caractéristique supplémentaire, il s'agit d'une modification en profondeur. elle modifie les automates du langage de définition de structure SOMINE.

La caractéristique Valeur + programme est une modification plus externe, elle ne crée qu'un type nouveau de mise en relation avec la base, elle agit au niveau des requêtes SOMINE.

Ces deux approches pourraient être regroupées en une caractéristique "PROCEDURE " qui du point de vue structurel serait de type Programme. Du point de vue de l'utilisateur on peut envisager plusieurs points d'entrée pour une Procédure, ceux-ci peuvent ou non être contenus dans la base (ils peuvent même appartenir à des bases différentes). Les points de sortie de la procédure seraient des variables SOMINE (à placer dans la base) ou bien des variables utilisables dans le programme qui déclenche l'exécution de la procédure. La procédure elle-même pourrait être un simple programme ou bien une chaîne de programmes contenant éventuellement d'autres accès à la base.

Elle peut être schématisée ainsi :



L'utilité d'une caractéristique aussi puissante est discutable surtout dans le cadre de cette application de SOMINE à la C.A.O. Il semble cependant que rien ne s'oppose à sa mise en place si une autre application de SOMINE la justifiait.

5 - REALISATION PRATIQUE DE L'APPLICATION

Le paragraphe précédent a donné plusieurs solutions au problème des abaques qui restait à résoudre.

SOMINE est maintenant un outil assez puissant pour nous permettre d'atteindre les buts que nous nous étions fixés en 2.2. à savoir pour les caractéristiques mécaniques des aciers :

- * Avoir une meilleure connaissance de certains paramètres et des corrélations qui existent entre eux.

- * Utiliser ces connaissances pour l'établissement d'une valeur acceptable du doublet (tenue en service, prix de revient) d'une structure métallique.

Avant de décrire les programmes qui vont permettre d'arriver à ces résultats, voyons qu'elles sont les modifications qu'il a fallu apporter au logiciel de base SOMINE pour le rendre utilisable pour cette application.

5. 1. - Mise en place de la caractéristique programme et valeur + programme au niveau de la structure des informations.

Les vecteurs qui représentent ces caractéristiques ont été décrits au paragraphe 4.3.3.

Dans la partie III, l'automate type élémentaire a déjà été décrit, les nouvelles caractéristiques lui donnent la forme suivante :

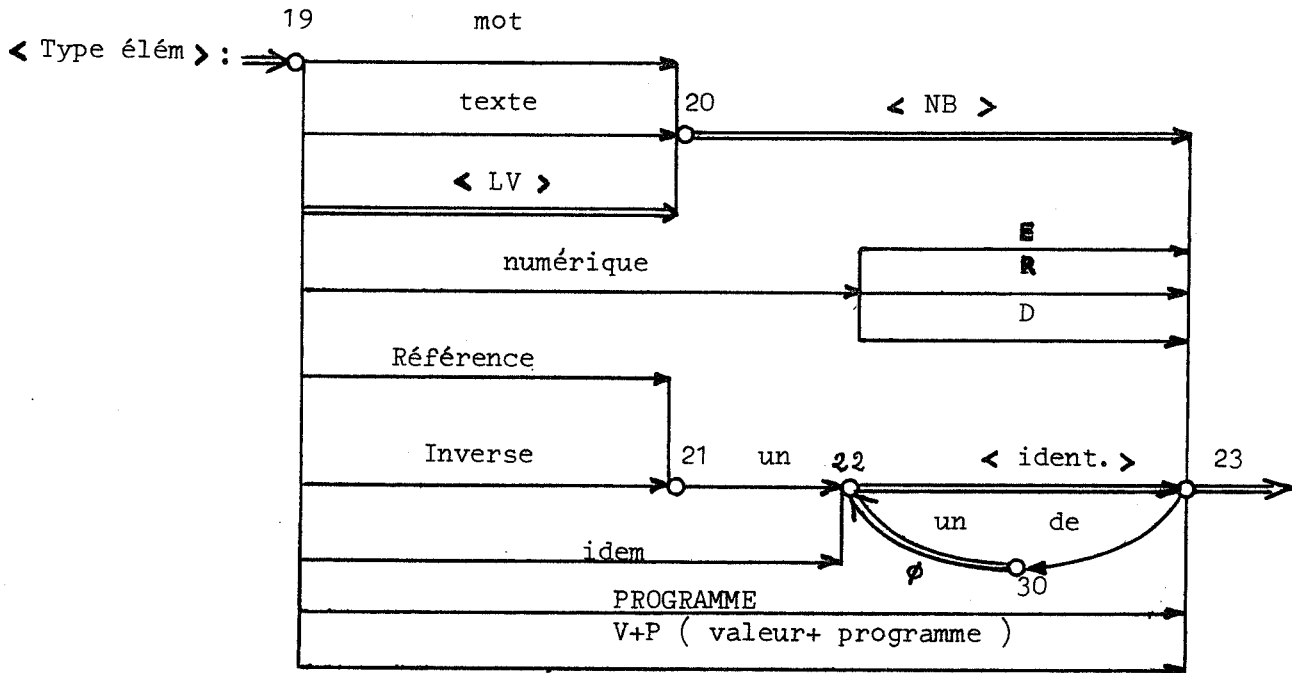


Figure IV.8

5. 2. - La mise en place des caractéristiques programme et valeur + programme du point de vue des requêtes.

Lors de l'analyse d'une requête dont l'élément de plus bas niveau est un programme nous avons vu qu'il y a appel du sous-programme d'aiguillage. Celui-ci en fonction du numéro structurel du programme et du numéro de réalisation des entités ancêtres, sélectionne le sous-programme à exécuter et déclenche son exécution.

Pour la caractéristique valeur + programme nous avons mis en place le mode supplémentaire VALEUR. L'exécution d'une telle requête commence par un test sur ce mode, il y a ensuite écriture ou lecture d'une caractéristique numérique, sélection et exécution d'un programme et rangement du résultat dans la banque.

5. 3. - La mise en place de la structure :

L'arbre de la structure retenue a été représenté au paragraphe 4. 1.

Nous allons maintenant écrire cette structure en L.D.S. ce qui indiquera le type et la désignation de chacune des caractéristiques.

PIECE

DEBUT

ENTITE 500 ESSAI

DEBUT

DATE MOT 10

LIEU TEXTE 1

NOM-MATERIAU MOT 8

TT NUMERIQUE E

COMPOSITION TEXTE 2

JOMINY TEXTE 1

R NU R

RE NU R

A% NU R

RR NU R

KCV20DEGRES NU R

KCU20DEGRES NU R

SIGMAM NU R

SOLLICITATION NU E

RT NU R

USINAGE NU E

KT NU R

SIGMAD NU R

SIGMASIGMAD NU R

FIN

METAUX TEXTE 4

STRUCTURE TEXTE 30

ENTITE 100 MATERIAU

.../...

DEBUT

ENTITE 5 TRAIT-THERMIQUE

DEBUT

SIGMA-1MAX NU R

SIGMA-1MIN NU R

SIGMA-1MOY NU R

RMAX NU R

RMIN NU R

RMOY NU R

REMAX NU R

REMIN NU R

REMOY NU R

FIN

FIN

ENTITE 3 MODSOL

DEBUT

KT PROGRAMME

EPSECH PROGRAMME

KPRINT PROGRAMME

ESPRIMECH PROGRAMME

FIN

ENTITE 10 MODE-DUSINAGE

DEBUT

EPSS PROGRAMME

EPSSPRIM PROGRAMME

FIN

FIN ***

Nous remarquons que la caractéristique de type texte désignée par STRUCTURE est destinée à contenir la structure en L.D.S. précédente. Ceci a été fait pour permettre l'écriture correcte des requêtes par un utilisateur néophyte ou inattentif.

La mise en place de la structure a donné l'image du fichier structure suivante :

.../...

DESIGN	CODE	MAX	PERE	OCC	TAILLE	FRERE	FILS	AR	AD		
PIECE	1	0	0	0	53349	0	15	0	0	-1	-1
ESSAI	3	500	0	0	95	315	30	0	0	-1	-1
DATE	9	10	15	0	3	45	0	1	0	-1	-1
LIEU	6	2	15	0	30	60	0	4	0	-1	-1
NOM-MATERIAU	9	8	15	0	2	75	0	34	0	-1	-1
TT	7	0	15	0	1	90	0	36	0	-1	-1
COMPOSITION	6	2	15	0	30	105	0	37	0	-1	-1
JOMINY	6	1	15	0	15	120	0	67	0	-1	-1
R	12	0	15	0	1	135	0	82	0	-1	-1
RE	12	0	15	0	1	150	0	83	0	-1	-1
AX	12	0	15	0	1	165	0	84	0	-1	-1
RR	12	0	15	0	1	180	0	85	0	-1	-1
KCU20DEGRFS	12	0	15	0	1	195	0	86	0	-1	-1
KCV20DEGRFS	12	0	15	0	1	210	0	87	0	-1	-1
SIGMAM	12	0	15	0	1	225	0	88	0	-1	-1
SOLLICITATION	7	0	15	0	1	240	0	89	0	-1	-1
RT	12	0	15	0	1	255	0	90	0	-1	-1
USINAGE	7	0	15	0	1	270	0	91	0	-1	-1
KT	12	0	15	0	1	285	0	92	0	-1	-1
SIGMAN	12	0	15	0	1	300	0	93	0	-1	-1
SIGMASIGMAN	12	0	15	0	1	0	0	94	0	-1	-1
METAUX	6	4	0	0	60	330	0	47517	0	-1	-1
STRUCTURE	6	30	0	0	450	345	0	47577	0	-1	-1
MATERIAU	3	100	0	0	53	510	360	48027	0	-1	-1
TRAIT-THERMIQUE	3	5	345	0	10	0	375	1	0	-1	-1
SIGMA-1MAX	12	0	360	0	1	390	0	1	0	-1	-1
SIGMA-1MIN	12	0	360	0	1	405	0	2	0	-1	-1
SIGMA-1MOY	12	0	360	0	1	420	0	3	0	-1	-1
RMAX	12	0	360	0	1	435	0	4	0	-1	-1
RMIN	12	0	360	0	1	450	0	5	0	-1	-1
RMOY	12	0	360	0	1	465	0	6	0	-1	-1
REMAX	12	0	360	0	1	480	0	7	0	-1	-1
REMIN	12	0	360	0	1	495	0	8	0	-1	-1
REMOY	12	0	360	0	1	0	0	9	0	-1	-1
MODSOL	3	3	0	0	1	585	525	53332	0	-1	-1
KT	15	0	510	0	0	540	0	1	0	-1	1
EPSECH	15	0	510	0	0	555	0	1	0	-1	2
KPRINT	15	0	510	0	0	570	0	1	0	-1	3
EPSPRIMECH	15	0	510	0	0	0	0	1	0	-1	4
MODE-DUSINAGE	3	10	0	0	1	0	600	53337	0	-1	-1
EPSS	15	0	585	0	0	615	0	1	0	-1	5
EPSSPRIM	15	0	585	0	0	0	0	1	0	-1	6
***	0	-2	-2	-2	-1	-1	-1	-1	0	-1	-1

5. 4. - Les données à mettre dans la base SOMINE :

* toutes les caractéristiques filles de l'entité ESSAI sont à remplir par l'utilisateur au fur et à mesure des résultats d'essais obtenus ou dépouillés. Nous verrons plus loin un programme permettant de réaliser cette mise à jour.

* les caractéristiques filles des entités MATERIAU, MODSOL et MODE DUSINAGE ont été dans un premier temps prises dans les bibliographies existantes. Nous allons les énumérer mais il convient de se rappeler que le but à atteindre est de prendre pour ces valeurs les résultats des programmes statistiques mis en route à chaque nouvel essai réalisé sur une éprouvette ou une structure existante.

données	numéro de référence bibliographique
$\sigma^{-1}(\sigma_b)$	
R	/1/ + /2/
R	
K_t	
K'_t	/5/
ξ_{ech}	
ξ'_{ech}	
ξ_s	/3/
ξ'_s	

5. 5. - Les données des programmes d'utilisation :

Les hypothèses du problème ont défini un certain nombre de matériaux, de traitements thermiques, de modes de sollicitation, et de modes d'usinage possibles, nous allons les énumérer pour mieux définir les hypothèses retenues.

.../...

- liste des matériaux :

E24		38C4	
E26		42C2	
E30		100C6	
E36	état	18CD4	état recuit + trempe
A50	recuit	25CD4	200°C
A60	unique-	35CD4	450°C
A70	ment	40CAD6-12	550°C
		42CD4	
XC10		30CD12	
XC12		50CV4	
XC18		10NC6	
XC25		14NC11	
XC38		16NC6	
XC42		20NC6	
XC48		30NC11	
XC65		16NCD13	
XC80		18NCD6	
XC100		20NCD2	
		35NCD6	
20M5		35NCD16	
16MC5		30CND8	
20MC5		13MF4	
45SCD6		35MF6	
55S7		Z8C17	
		Z30C13	

- Les traitements thermiques :

recuit

trempé revenu température de détente 200°

trempé revenu température de détente 450°

trempé revenu température de détente 550°

- les modes de sollicitations :

les trois modes de base sont la flexion
la traction
la torsion

Ces trois modes peuvent être combinés entre eux ce qui donne :

$$C_3^3 + C_3^2 + C_3^1 = 7 \text{ modes de sollicitations possibles.}$$

- les modes d'usinage :

fraisé
rectifié
tourné
rodé ou poli
brut de laminage ou brut de forge

Le programme utilisateur aura besoin d'un certain nombre de données pour faire les recherches documentaires et les calculs nécessaires. On distingue quatre groupes de données dont les notations et la signification ont déjà été données (§2.4.2.)

- Celles qui sont relatives à la forme de la pièce et à celle de l'entaille.

a rayon de la pièce au fond de l'entaille
r rayon de l'entaille (voir 2.4.3.)
t profondeur de l'entaille

forme de la pièce : plate ou cylindrique

- Celles relatives à l'état de contrainte:

σ_{dyn}	τ_{dyn}
σ_{stat}	τ_{stat}
mode de sollicitation	

- Celles relatives à l'état de surface.

R_t
mode d'usinage

.../...

- Celles relatives au matériau

nom du matériau

traitement thermique

5. 6. Le programme d'aiguillage

Il sert à la sélection du programme à exécuter au moment de l'appel d'une caractéristique PROGRAMME.

Ce programme pourrait être écrit une fois pour toute mais alors il imposerait des limites au nombre et au niveau hiérarchique des programmes susceptibles d'être appelés. Le résultat serait une absence de souplesse et un encombrement mémoire dû à la taille du programme d'aiguillage.

En effet, si nous nous limitons à dix sous-programmes à appeler, chacun pouvant être situé au niveau hiérarchique 6 et si les entités de ces niveaux ont seulement dix réalisations on obtient :

$$10 \times 10^6 = 10^7 \text{ sous-programmes !}$$

Nous voyons donc que trois facteurs interviennent dans le programme d'aiguillage, ce sont :

- le nombre de programmes
- le niveau de ces programmes
- le nombre de réalisations possibles à chaque niveau

L'influence des deux derniers facteurs est très grande sur le nombre de programmes qui devront être appelés à partir de la base.

Il semble donc qu'il vaille mieux laisser à celui qui implémente une base le soin de rédiger son programme d'aiguillage, il est le seul qui connaisse l'importance des trois facteurs cités plus haut, il pourra ainsi optimiser la taille de ce programme.

Une autre façon d'envisager l'aiguillage est la création d'une table de correspondance entre les noms des programmes et les numeros structurels. La recherche est simplifiée tant que le nombre de programmes n'augmente pas trop le temps de recherche .

6 - LES PROGRAMMES AYANT ACCES A LA BASE SOMINE CREEE

Dans la partie **III**, lorsque nous avons mis en place les moyens d'accès à une base SOMINE, nous avons voulu qu'ils soient souples, simples et variés, En particulier conversationnels. /19/

Ces particularités ajoutées au fait que le logiciel SOMINE peut être appelé à l'intérieur de tout programme et que ce même logiciel peut déclencher l'activation de programmes (caractéristiques programme) vont nous permettre de montrer que SOMINE peut être considéré comme le centre d'un système d'aide à la conception des projets scientifiques dans le domaine des caractéristiques mécaniques des matériaux.

6. 1. - Schéma général :

La base SOMINE que nous venons de décrire ainsi que les programmes nécessaires à l'application que nous traitons sont schématisés par la figure IV.9.

La base peut donc être considérée comme la réunion de deux sous-bases bien distinctes. Ces deux sous-bases se différencient par leur mode de mise à jour et l'utilisation que l'on va pouvoir en faire nous allons les résumer maintenant en faisant référence aux numeros portés sur le schéma de la page suivante.

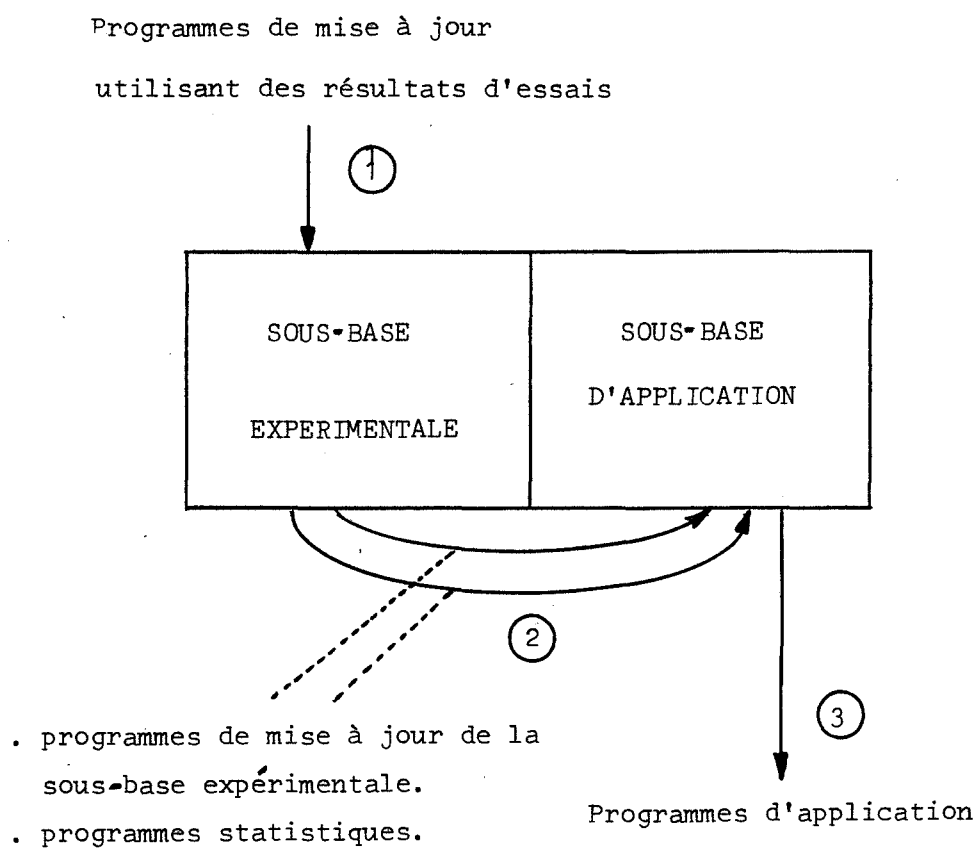


Figure IV.9

Sous-base expérimentale :

* Remplie par l'utilisateur d'après les résultats des essais (1)

* Chaque essai active des programmes. Ceux-ci cherchent les plages de valeurs possibles pour les données numériques (ainsi que les valeurs moyennes et extrémales), d'autres programmes de correlations statistiques permettent d'approcher de plus en plus les relations qui existent entre les éléments de la sous-base expérimentale (2)

Sous-base d'application :

* Remplie par les résultats des programmes précédents (2)

* Les informations qu'elle contient servent à des applications scientifiques (3).

Ces différents types de programmes peuvent être des programmes existants dans le cas des programmes statistiques, d'autres ont été écrits pour les besoins de l'application. Nous allons énumérer les programmes entrepris mais tout nouveau traitement des données de la base peut toujours donner lieu à un nouveau programme, le système d'aide à la conception n'est pas un système fermé et une grande initiative est laissée à l'utilisateur pour lui permettre de traiter les problèmes qu'il veut dans l'ordre de leur priorité.

6. 2. - Programme de mise à jour de la sous-base expérimentale - programmes statistiques.

Pour remplir les fonctions (1) et (2) du schéma n° 9 nous avons écrit un programme ESSAI dont le listing sera trouvé en annexe à ce travail.

C'est un programme FORTRAN conversationnel dont les deux fonctions essentielles sont :

- la mise à jour de la sous-base expérimentale à l'aide de données collectées sur un écran alphanumérique

- la mise à jour de la sous-base d'application avec les valeurs moyennes et extrémales des caractéristiques collectées à l'écran

et déjà contenues dans la base (limite élastique, limite de fatigue, résistance à la rupture)

- ce programme sera complété ultérieurement par des appels à des programmes FORTRAN permettant de faire des interpolations d'abaques ou d'établir des correlations entre différents paramètres contenus dans la base. Les travaux se poursuivent en ce moment dans ce sens là en liaison avec le CETIM.

6. 3. - Calcul du coefficient de sécurité en mode conversationnel

L'avantage de cette méthode est le dialogue qui s'engage entre la base le programme et l'utilisateur. Les trains de données nécessaires à son exécution peuvent être demandées dans l'ordre voulu et au moment où elles sont nécessaires ; de plus, l'accès au résultat est immédiat. Des séries d'essais peuvent être envisagées en modifiant tout ou une partie seulement des données d'entrée. Ce programme de calcul permet ainsi une approche de la solution optimale qui peut être basée sur la valeur du coefficient de sécurité ou sur le prix de revient du projet conçu.

Dans le cadre de ce projet, la mise en oeuvre de ce programme est achevée malgré le problème que pose l'introduction des données numériques par écran. En effet, tout ce qui est écrit sur l'écran est considéré comme une chaîne alphanumérique. Il est donc nécessaire d'utiliser un fichier auxiliaire pour faire les conversions et les différents accès à ce fichier allongent le temps d'exécution. Un autre intérêt de cette méthode est la possibilité de conserver un document écrit après une session en couplant un écran avec un télétype.

Le listing de ce programme sera trouvé en annexe de ce travail mais nous allons énumérer les différentes fonctions de ce programme :

- + c'est un programme FORTRAN conversationnel,
- + il prend un certain nombre de données sur l'écran (données numériques réelles ou entières et alphanumériques)
- + il va chercher d'autres données dans la sous-base d'application

+ il active des caractéristiques programme et recueille les arguments de sortie de ces programmes.

+ il effectue du calcul scientifique avec toutes ces données

+ il peut faire plusieurs essais successifs en faisant varier les paramètres donnés à l'écran.

6. 4. - Calcul du coefficient de sécurité en traitement par lots de programmes.

Un programme remplissant les mêmes fonctions et permettant de calculer le coefficient de sécurité α suivant le même algorithme que le précédent a été écrit. Ce programme une fois rangé sur une bibliothèque disque peut être activé et demande seulement la connaissance des jeux de données nécessaires aux calculs envisagés. Ce mode de calcul est moins souple mais a l'avantage de pouvoir s'effectuer sans la présence de la personne intéressée puisqu'il n'est pas conversationnel.

6. 5. - Un exemple d'action spontanée :

Plusieurs programmes ont été considérés comme des caractéristiques de la base SOMINE dont nous expliquons la mise en place. Parmi eux il y a le sous-programme KT permettant de calculer K_t le coefficient de concentration de contraintes statiques en fond d'entaille en sollicitations normales. Il présente la particularité de faire lui-même appel à la base.

Un extrait du listing se trouve à la page suivante .

```

100      X(1)=ITT
        X(2)=IMAT
        WRITE(3,300)ITT,IMAT,NUCIT(2)
300      FORMAT(1X,'ITT=',15,10X,'IMAT=',15,10X,'NUCIT(2)=',15)
        CALL INTER (' I RMOY DU TRAIT-THERMIQUE X(1) DU MATERIAU X(2) #')

C          CALCUL DE Q CONNAISSANT R ET KT
        IF(AT.LT.1.4) GO TO 101
        IF(RMOY.GE.1200.)GOTO101
        IF(RMOY.GE.1100.)GOTO103
        IF(RMOY.GE.900.)GOTO104
        IF(RMOY.LT.600..AND.RMOY.GT.400.)GOTO105
        IF(RMOY.LT.400.)GOTO106
        Q=0.75
        RETURN
101      Q=1.
        RETURN
103      Q=0.9
        RETURN
104      Q=0.8
        RETURN

```

Nous remarquons que le calcul de K_t (ici l'identificateur est AT) et la recherche de R (ici RMOY) (résistance à la rupture) dans la base permettent de calculer la valeur de q (sensibilité à l'entaille).

7 - RESULTATS ET CONCLUSION

Nous avons dans ce rapport tenté une approche informatique d'un problème d'aide à la C.A.O. qui n'avait jamais été résolu auparavant.

A savoir la réalisation et l'utilisation d'une base de données contenant les caractéristiques mécaniques des aciers.

L'aspect réalisation de cette base SOMINE a permis une réflexion et une adaptation des bases de ce type au problème considéré. Ceci a permis d'élargir la définition de **SOMINE** et de créer les caractéristiques programme, valeur + programme et enfin procédure. Leur but est de permettre l'activation de programme à partir d'un accès à la base. Cette possibilité prend tout son intérêt lors de sessions à l'écran, l'utilisateur peut ainsi par le biais d'un accès à SOMINE faire exécuter des programmes et reprendre ensuite son travail en utilisant (ou non) les résultats de ce programme.

SOMINE est maintenant un système de base de données dans lequel :

Les accès aux bases peuvent se faire à l'intérieur de tout programme écrit en langage algorithme

Les accès aux bases peuvent activer des programmes écrits dans ces mêmes langages.

L'aspect utilisation de la base SOMINE créée a permis une réflexion sur la nécessité, la mise à jour d'une telle base et les buts recherchés ou pouvant être obtenus.

.../...

Le schéma du paragraphe 6. 1 , résume les deux aspects de la base créée. Une partie de cette base (sous-base expérimentale) contient les résultats de tous les essais et expériences réalisés. Ces résultats expérimentaux sont ainsi conservés dans un but d'archives, de centre de documentation toujours disponible. Actuellement cette documentation est écrite et aurait besoin d'être rassemblée et dépouillée dans le but de dégager des valeurs moyennes, des écarts types de caractéristiques des aciers..... dans le but de juger de la vraisemblance ou la compatibilité des résultats expérimentaux avec les expériences ayant déjà eu lieu. Tous ces buts peuvent être atteints par des programmes statistiques ou de comparaison. Cette sous-base expérimentale outre son aspect "centre de documentation" pourra alimenter les programmes précédents... mais alors où allons nous mettre les résultats de ces programmes : valeurs moyennes, relations définies... mais dans la base SOMINE. Cette fois-ci nous utiliserons la partie appelée (sous-base d'application) elle contiendra à tout moment le point de recherches sur les valeurs des caractéristiques des aciers et servira d'aliment aux programmes d'application utilisant des données.

La réalisation de la base, la mise à jour des deux sous-base ont été effectuées. Des programmes de calcul de moyennes, de valeurs extrémales sont au point, l'établissement des corrélations statistiques demande une étude plus approfondie et la documentation nécessaire n'est pas encore dépouillée totalement. Des travaux sont poursuivis au CETIM dans ce but là, pour notre part, en utilisant les résultats connus nous avons essayé d'appliquer les connaissances contenues dans la base à l'établissement des lois de comportement des structures mécaniques.

Dès à présent, la série de solutions proposées permet d'atteindre les objectifs que nous nous étions fixés c'est à dire dans le cas de pièces qui présentent actuellement un taux de défaillance important diminuer le nombre des essais de simulation en général coûteux et permettre d'optimiser le choix des paramètres technologiques en fonction d'un coefficient de sécurité et d'un prix de revient.

Cette notion de prix de revient est calculée dès à présent pour une solution donnée en introduisant dans la base (le prix du matériau le coût de la mise en forme, celui des traitements et des usinages). Elle pourrait être introduite ultérieurement dans la base à l'occasion

d'un traitement d'optimisation automatique, lorsque les diverses études menées dans des centres de recherche industriels et en particulier au CETIM sur les temps d'usinage et les familles de pièces auront abouti.

La résolution du problème énoncé dans ce rapport se place dans un système de C.A.O. plus vaste et déjà en partie informatisé. Il semble que dans un premier temps la base statistique permette par son mode d'utilisation en conversationnel de cerner assez rapidement des solutions acceptables pour le problème posé. Dans un deuxième temps le calcul de l'état de contraintes réel (par la méthode des éléments finis ou des équations intégrales par exemple) permettra un travail plus précis

En ce sens nous pensons avoir ouvert une voie qui une fois améliorée ou reprise pourra donner naissance à la solution totalement informatisée du problème général de la tenue en service d'une structure. Il faudrait pour cela que les recherches en cours sur les critères d'endommagement des matériaux aboutissent dans des cas de sollicitations plus complexes que celles envisagées dans ce texte.

Ce système de calcul de la tenue en service d'une structure pourra d'ailleurs être raccordé en amont aux problèmes de leurs définitions graphiques, définitions qui mobilisent actuellement des équipes importantes.

Or une solution automatisée est envisageable car il existe du matériel informatique pour faire du dessin automatique ou semi-automatique /7/, /15/. L'impossibilité d'avoir accès à ce matériel est la raison pour laquelle nous n'avons pas poussé notre réflexion dans ce domaine. Cependant un système de base de type **SOMINE** pourrait contenir une banque graphique (éléments standardisés de dessins) et le couplage de cette base graphique avec une table à dessin ou un écran graphique aurait un double avantage. D'une part au moment de la modélisation et d'autre part au moment du tracé définitif lorsque le problème est résolu. De plus ce couplage permet de faire effectuer à l'ordinateur des fonctions, l'interpolation par exemple, permettant de passer ainsi directement du tracé de la courbe à son expression algébrique approchée. Un exemple de tel couplage ordinateur-machine à dessiner est donné par le système COMPAC /14/.

On pourra ainsi tenter de résoudre le problème de l'optimisation de la solution générale de la conception d'une structure destinée à une utilisation déterminée. Il semble au premier abord que seul un spécialiste ayant une "expérience du système à concevoir" soit à même de trouver ou non une solution satisfaisante, d'où l'intérêt du mode conversationnel. Ne pourrait-on pas envisager de laisser à l'ordinateur l'appréciation de l'optimisation de la solution ? Cette méthode présenterait un très grand intérêt mais naturellement le temps d'exécution machine serait sans doute plus long, car "l'expérience du projeteur" serait remplacée par une série d'essais. Ainsi le projet n'aurait pas à être prédéterminé à l'avance. Pour cela la base devrait avoir connaissance des solutions aberrantes du point de vue technologique ou prix de revient. Une nouvelle voie semble ouverte dans ce sens et nous croyons que l'analyse de ce problème permettra un morcellement des éléments d'appréciation de la solution optimale et d'en définir alors une meilleure formalisation.

Nous pensons avoir montré l'importance de l'expérimentation en informatique et la façon dont elle permet "la conception ascendante" c'est-à-dire l'amélioration d'un outil de base dans le but d'atteindre un objectif fixé.

La souplesse et la simplicité d'utilisation des bases de type SOMINE ont permis d'associer dans un même programme des problèmes numériques, des problèmes documentaires et de donner une solution au problème des actions implicites qui garantissent la cohérence de la base.

Nous remercions le CETIM de Saint-Etienne qui a bien voulu nous faire confiance et nous permettre d'avoir accès à sa documentation. C'est la compétence en résistance des matériaux de ses chercheurs qui a permis à ce projet de voir le jour tant il est vrai que "la réalisation d'un traitement de l'information dans une activité humaine nécessite la connaissance de ce qui est mis en jeu au plan sémantique dans cette activité" /12/

Nous remercions également Mr Brun (L.I.M.S.I. Orsay) et Mr Bahuaud (I.N.S.A. Lyon) qui ont bien voulu lire ce texte et nous apporter leurs remarques et leurs suggestions .

Nous espérons dans cette partie avoir mis en évidence les possibilités et les originalités de SOMINE et avoir ainsi apporté notre contribution dans le domaine de la C.A.O. Notre vœu est que les méthodes que nous avons mises à jour soient complétées ou reprises ou bien qu'elles donnent naissance à de nouveaux développements dans ce domaine tant il est vrai que l'informatique industrielle plus d'ailleurs que l'informatique de gestion traditionnelle n'est une fin en soi. Chacune d'elles n'est qu'une étape vers une information totalement intégrée de l'entreprise...

/21/. /20/

8 - BIBLIOGRAPHIE DE LA PARTIE IV

- /1/ - R. CAZAUD, G. POMEY, P. RABBE, CH. JANSEN - "La fatigue des métaux"
DUNOD
- /2/ - Publications scientifiques et techniques du ministère de l'air.
Caractéristiques des aciers - 1958
- /3/ - BAHUAUD, BOIVIN, RUME LHART et VASSAL; "le calcul des structures
soumises à la fatigue" , recueil des conférences I.N.S.A.-Lyon 1971
- /4/ - E. HOTTENROTT, Ingénieur de recherche à l'Office National d'Etudes et
de recherches Aéronautiques, "contribution au calcul pratique des
facteurs de concentration des contraintes"
- /5/ - R. E. PETERSON, "Stress Concentration design factors" - John Wiley &
Sons, inc, NEW-YORK
- /6/ - Ecole Nationale Supérieure des techniques avancées (E.N.S.T.A.)
système C.A.O. : " réalisation et structure d'une bibliothèque de
sous-programmes techniques pour utilisation en mode conversationnel
par un bureau d'étude mécanique" - projet n° 34/E - 1974
- /7/ - Méthodes graphiques en mécanique - "Journées organisées par le
G.A.M.I. et l'A.D.E.P.A. sous le patronage de l'A.F.C.E.T. et de la
S.I.C.F. - 6 et 7 juin 1974 à l'E.N.S.T.A.
- /8/ - J. ARSAC - "Sorbon"
- /9/ - J.C. LATOMBE et J.C. SABONNADIÈRE - "Elaboration d'un système infor-
matique intégré pour l'assistance à l'enseignement et à la conception
en électrotechnique "
- /10/ - DERMIANE, Université de Nancy I - Projet CIVA - "un système de
programmation modulaire" - thèse d'Etat - janvier 1974
- /11/ - WIRTH , program developpement by stepwise refinement C.A.C.M.
14,4 - April 1971
- /12/ - J. ARSAC, " La science informatique" DUNOD

- /13/ - BRACCHI et SOMALVICO - "An interactive software for computer aided design : an application to circuit projet". communication of the A.C.M. - vol 13 - n° 9 - septembre 1970
- /14/ - P. BEZIER et G. LETAC : " traceurs de courbes et machines à dessiner à commande numérique".
P. SPUR et H. DEBLER - " compac : a system for computer aided design of mechanical workpieces"
Mécanique, matériaux électricité - organe officiel du G.A.M.I. - N° 300 Sciences et industrie - décembre 1974
- /15/ - L. ETESSE - R. PRAJOUX, D. ESTEVE, laboratoire d'automatique et d'analyse des systèmes du C.N.R.S. , 7 avenue du Colonel roche 31400 TOULOUSE, " problèmes généraux liés à l'automatisation intégrée d'unités de production" colloque E.X.E.R.A. - Paris 5 février 1975.
- /16/ - L. SORS "Fatigue design of machine components-" Pergamon Press
- /17/ - P. MARTY thèse de doctorat de troisième cycle, Ecole Nationale Supérieure des Mines de Saint-Etienne, " Contribution à la conception, à la réalisation et à l'utilisation du système de gestion de bases de données SOMINE : Structuration des informations." (1976
- /18/ - M. GAILLARD, thèse de doctorat de troisième cycle, Ecole Nationale Supérieure des Mines de Saint-Etienne, " Contribution à la conception, à la réalisation et à l'utilisation du système de gestion de bases de données SOMINE : gestion des mémoires - Enseignement Assisté par Ordinateur. " 1976
- /19/ - J.J. GIRARDOT " utilisation des écrans sous SYST DC"- info info de l'Ecole Nationale Supérieure des Mines de St.Etienne - février mars 1973
- /20/ - M. ESTEVE , maître de conférences au LAAS de Toulouse, "l'automatisation intégrée dans les systèmes industriels" - bulletin de liaison de la recherche en informatique et automatique" n° 13 février 75
- /21/ - M. ALLEGRE, " réflexions sur l'informatique industrielle" février 72
- /22/ - R. WEILL, J.P. GILLET, M. VANCHELEMONTE, M. BARAT, JC PASQUIES "études des caractéristiques optimales d'un système de banques de données de coupe appliquées au tournage en commande numérique"

PARTIE V
CONCLUSION

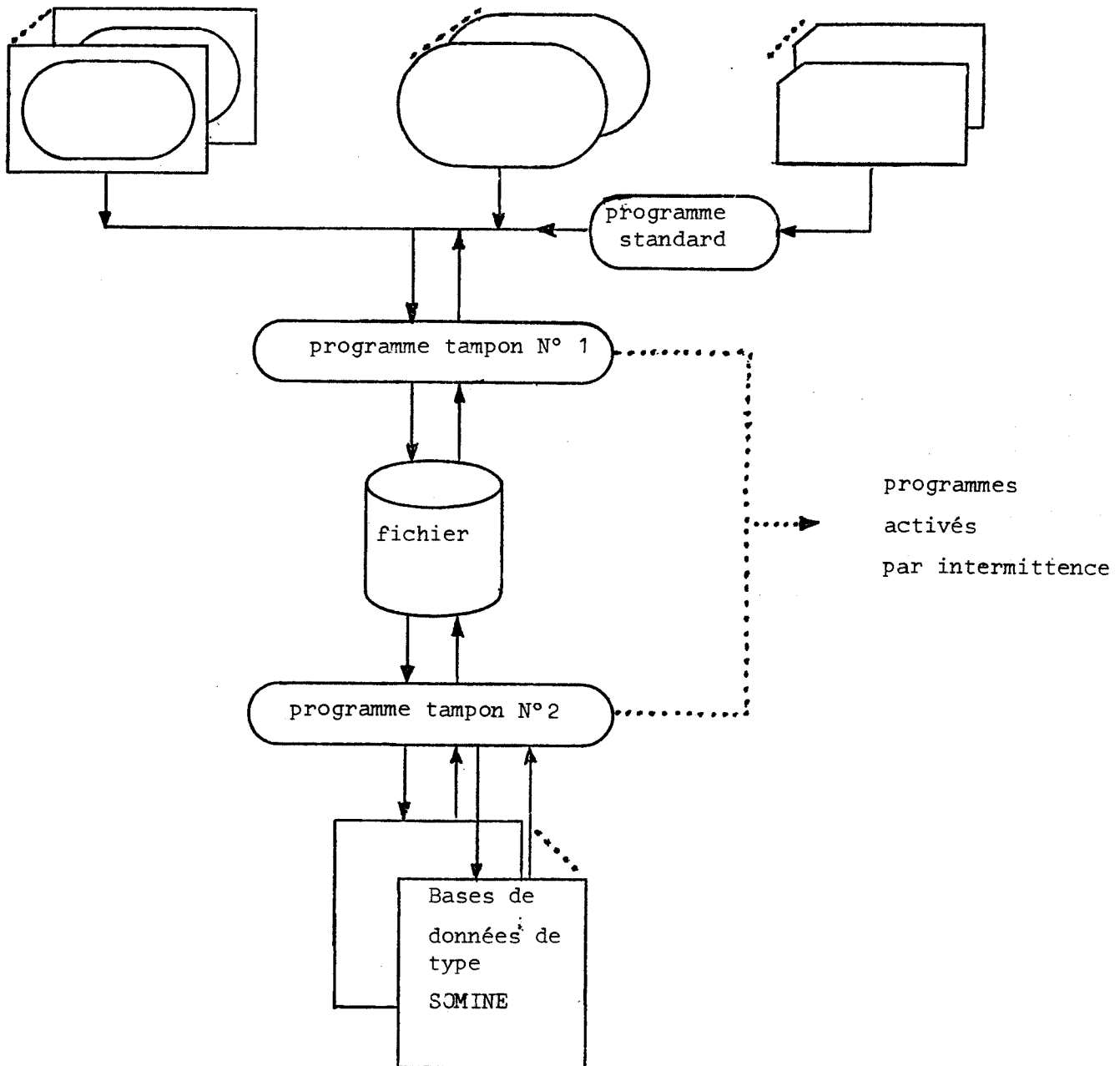
Voici venu le moment de faire la synthèse et le bilan des travaux que nous avons effectués et dont une partie fait l'objet de cette thèse. C'est le système de bases de données SOMINE qui constitue l'essentiel de nos travaux, nous allons le schématiser et le concrétiser avec les figures des deux pages suivantes. La première représente l'utilisation d'un système de bases de données de type SOMINE, la deuxième illustre le fonctionnement d'une de ces bases depuis l'étape de création jusqu'aux différentes possibilités d'utilisation.

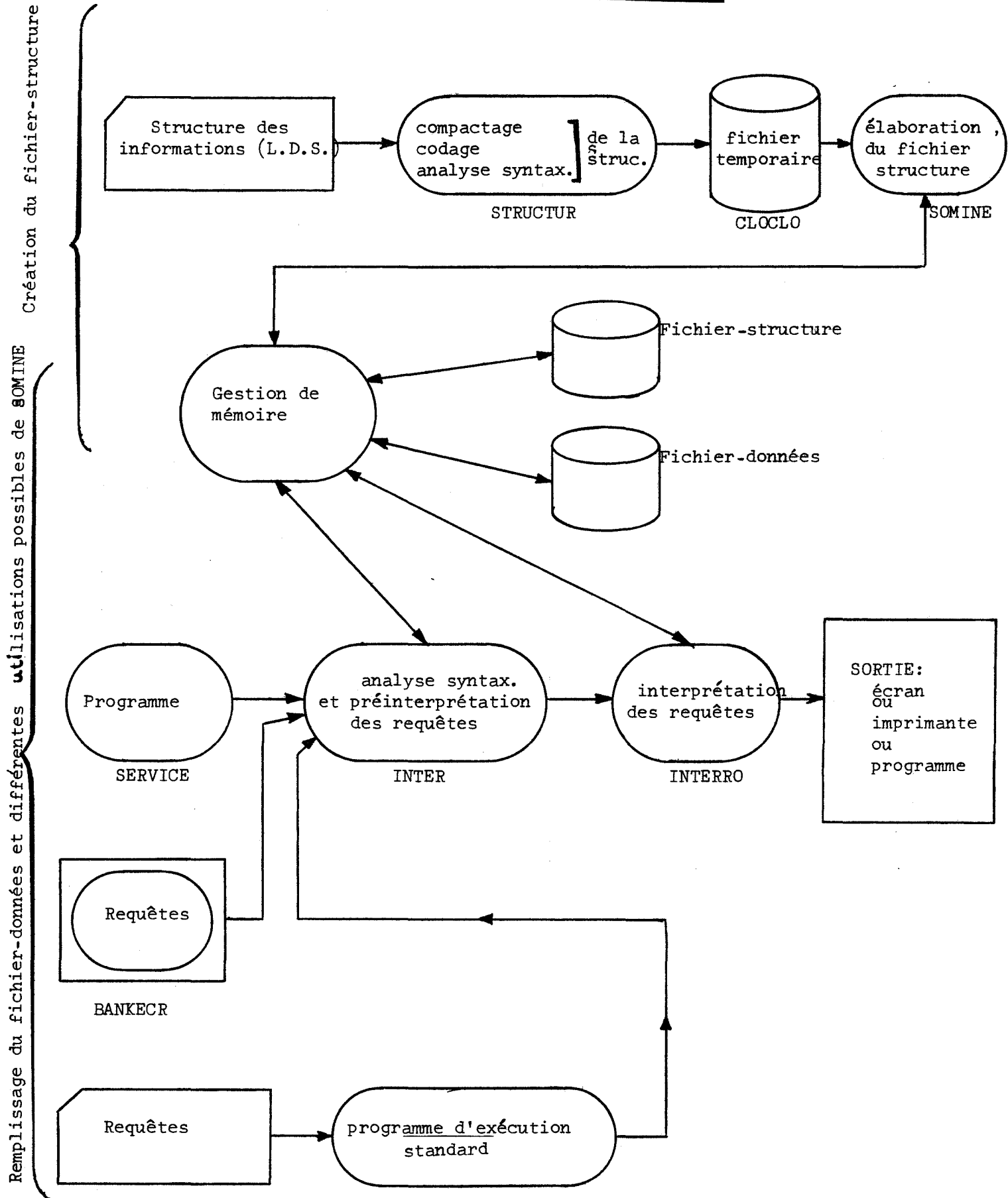
SYSTEME DE BASE DE DONNEES SOMINE

écrans

programmes d'utilisation

requêtes sur des cartes



UNE BASE DE DONNEES SOMINE

En réalisant le système de bases de données SOMINE, nous avons essayé de résoudre le problème de l'optimisation de l'organisation et de la recherche des informations appartenant à une collection d'informations.

Par certains côtés ce souci de discipline dans l'organisation et la structuration des informations en vue d'approcher d'une connaissance n'est pas sans rappeler la structure même du cerveau humain. On peut aussi penser aux philosophies orientales qui par des postures physiques et mentales essayent d'atteindre la connaissance et le bonheur.

Nous nous sommes donc attachés à trouver une philosophie applicable aux bases de données et assurant la vie de la collection d'informations qu'elles gèrent. Cette vie devra être conforme aux objectifs que nous nous étions fixés dans l'introduction de ce rapport, la nécessité de quelques autres critères s'est aussi dégagée au cours de notre travail.

Le premier objectif retenu est essentiel puisque c'est de lui que découle la notion même de base de données. Il s'agit d'assurer un traitement varié et efficace des informations contenues dans la base. Nous pouvons développer un peu les principales actions qui devront être réalisables pour que le traitement des informations soit satisfaisant. La base devra assurer :

- le placement des informations dans les mémoires auxiliaires réservées à cet effet.
- la recherche et la découverte rapide d'une information sur ces mêmes supports.
- enfin, et c'est ce point qui permet de juger de l'efficacité d'une base, les informations doivent pouvoir être manipulées dans le but d'en créer de nouvelles. Ces créations peuvent être faites de plusieurs façons, nous pouvons citer les calculs sur les éléments de la base ainsi que les informations composées par coordination ou subordination des éléments de la base. Enfin des applications telles que la concaténation d'informations ou les filtres sur les informations permettent de juger de l'aptitude à donner naissance à de nouvelles informations obtenues à partir de celles qui sont dans la base.

.../...

Si le traitement des informations est assuré par une base, on peut se fixer d'autres objectifs permettant d'en faciliter l'utilisation ou d'en augmenter les qualités. les critères suivants nous ont semblés intéressants à retenir :

- indépendance du système par rapport au matériel utilisé ceci afin d'en assurer la transportabilité.

- indépendance de la collection d'informations par rapport aux utilisateurs (personnes ou programmes)

- adaptabilité à de nouvelles exigences .

- fiabilité et protection de la base .

- mesurabilité du système en vue d'améliorer ses performances ou de modifier la structure des informations.

Ces critères que nous avons retenus sont le fruit de notre réflexion et de notre expérimentation . Nous avons également consulté " l'étude sur les softwares de gestion et d'interrogation des fichiers " publication éditée par la délégation à l'informatique groupe " systèmes d'informations " (avril 1973).

Ce document constitue une très bonne étude comparative des différentes bases de données. Il se termine en donnant pour chacun des systèmes de bases de données étudié une fiche signalétique. Nous avons retenu ce modèle de fiche signalétique pour SOMINE, nous l'avons complété et placé en annexe de cette conclusion.

Le lecteur pourra s'y reporter pour connaître des détails techniques mais nous allons faire ici un bref rappel des originalités ou des qualités du produit SOMINE; dans cette énumération nous aurons présent à l'esprit les objectifs que nous nous étions fixés.

.../...

- Naturellement les bases SOMINE peuvent effectuer les actions de rangement et de recherche des informations qui leur sont confiées. Nous avons vu dans la partie III que ceci est possible grâce aux requêtes de base dont la forme est simple et logique et dont l'apprentissage est rapide.

- La mise en oeuvre des filtres (caractéristique inverse) et des requêtes booléennes assurent une bonne faculté de création d'informations non contenues explicitement dans la base. Toute requête contient une citation hiérarchique elle assure donc la naissance d'informations composées par subordination, quant aux informations composées par coordination elles peuvent être obtenues par les requêtes groupées ou celles contenant le séparateur "tout". Une assez grande possibilité de manipulation des informations est laissée à l'utilisateur qui pourra les mettre en oeuvre dans ses programmes d'utilisation.

- Le système peut être amélioré assez facilement car la programmation des différentes parties est modulaire. De plus les automates de définition sont simples et une ou plusieurs transitions entre deux états sont toujours possibles à programmer. En somme nous pouvons dire que SOMINE est un produit évolutif, c'est -à- dire que le rajout d'une nouvelle fonction de manipulation, d'un nouveau type de caractéristique ou d'un nouveau mode de recherche ne brise en rien l'unité du système. On en trouvera une illustration dans la partie IV de ce rapport.

- Puisque ce logiciel est adaptatif il est perfectible. Naturellement la programmation n'en est pas parfaite et une optimisation peut toujours être envisagée, mais le plus intéressant est la possibilité d'effectuer une programmation plus fine (en assembleur par exemple) de certaines parties du système. Ceci ne peut être envisagé nous semble -t-il que dans le cadre d'une application donnée et pour un matériel donné. En effet généraliser cette remarque irait à l'encontre de l'objectif suivant qui est :

- l'indépendance vis-à-vis du matériel utilisé. Pour Somine celle-ci est assurée par la programmation effectuée presque totalement en Fortran d'où la transportabilité du produit.

- rappelons que les programmes d'implémentation de la structure et les programmes d'utilisation et d'analyse des requêtes sont totalement indépendants, ceci évite d'avoir à recréer le fichier des informations si les traitements qu'elles subissent varient.

- SOMINE est un système mesurable sous deux aspects. D'une part l'évaluation du temps d'exécution de l'implémentation ou de l'utilisation d'une base peut être faite facilement. D'autre part des compteurs situés dans la structure permettent d'espionner et de compter le nombre d'accès à chaque caractéristique. Une simple lecture de ces compteurs permet des évaluations statistiques et peut-être une modification de la structure des informations s'en déduira ce qui pourra contribuer à l'optimiser.

- du point de vue de l'utilisateur nous remarquons que les modes d'accès aux bases de type SOMINE ont été diversifiés : batch, écran, programme. Ce dernier point, l'appel par programme, est dû au fait que les requêtes sont traitées comme des chaînes alphanumériques et peuvent donc faire partie d'un programme écrit dans un langage qui permet leur traitement. Ces chaînes alphanumériques sont des phrases syntaxiquement correcte du langage de requête dont l'apprentissage est particulièrement rapide et logique.

- la protection peut être assurée facilement par un mot (ou un code) de passe contenu dans la requête ou la précédant.

- l'utilisation simultanée d'une base par plusieurs utilisateurs est réalisée par deux programmes tampons activés à des intervalles de temps réguliers et qui ont accès à un fichier qui contient les requêtes ainsi que le nom de l'utilisateur, ce fichier sert aussi de véhicule aux informations qui transitent entre la base et les utilisateurs. Ces programmes et ce fichier assurent le partage du temps entre les différents utilisateurs, les accès à la base sont déclenchés dans l'ordre dans lequel ils ont été demandés.

Ces différents rappels ont permis de montrer que SOMINE a atteint les objectifs que nous avons retenus dans l'introduction.

On peut se demander si SOMINE n'est pas un langage. Nous pensons que oui et qu'il est complémentaire des langages algorithmiques comme Fortran, Algol, PL/1 puisque les requêtes de SOMINE peuvent être traitées par ces langages. Nous ne pensons pas que SOMINE fasse double emploi avec PL/1 car la définition des structures et les différentes utilisations possibles sont dynamiques dans SOMINE. Comme atout pour notre produit nous pouvons ajouter le fait que la gestion de mémoire est tout à fait invisible à l'utilisateur. SOMINE a aussi l'avantage d'être efficace dans la recherche des données manipulées.

.../...

En effet, la structuration des informations sous forme de graphe ou d'arbre est plus riche que la plupart des structurations trouvées dans les langages de programmations usuels. Rappelons que :

- Fortran travaille sur des tableaux de taille fixe.
- Algol travaille sur des tableaux de tailles variables
(la mémoire libre est gérée par le système)
- Dans PL/1 il existe des structures imbriquées mais l'allocation et la désallocation de la mémoire doivent être demandées par l'utilisateur.
- Dans le langage Pascal il existe également des structures mais elles sont statiques.

Nous espérons que ces rappels et ces comparaisons ont mis en évidence les originalités de SOMINE. La souplesse et la simplicité des accès à la base favoriseront, nous l'espérons, son utilisation dans divers domaines.

Nous allons faire maintenant la synthèse de l'aspect utilisation de l'outil SOMINE. Nos recherches se sont orientées dans trois domaines qui nous ont semblés susceptibles de nécessiter l'utilisation d'une base de données et de mettre en évidence les qualités de notre produit.

* Domaine de l'E.A.O. (enseignement assisté par ordinateur) voir thèse de M. GAILLARD.

* Domaine de l'étude des structures voir thèse de P. MARTY

* Dans ce rapport nous nous sommes plus particulièrement penchés sur le domaine de l'aide à la C.A.O. (conception assistée par ordinateur). Nous allons rappeler les résultats que nous avons obtenus et les perspectives d'avenir qui en découlent.

Cette partie commence par une analyse des différents types de problèmes que pose la conception en général. Certains de ces problèmes (résolutions numériques, simulations) reçoivent parfois une solution informatique, nous avons essayé de mettre en évidence le bénéfice que pourrait apporter le traitement d'un plus grand nombre de ces problèmes par cette méthode (en particulier dans la recherche documentaire et les résolutions graphiques). Cet essai d'informatisation de la conception d'un projet scientifique ou technique nécessite la mise en mémoire d'un grand nombre d'informations de types variés

(valeurs numériques, programmes, portions de dessins standardisés....) C'est à ce niveau qu'une base de données efficaces et suffisamment simples pour ne pas rebuter les concepteurs peut être utilisée avec profit.

La possibilité d'exécuter simultanément de la recherche documentaire et toutes les autres instructions des langages algorithmiques fait de SOMINE une base adaptée à la C.A.O. Pour notre part nous avons envisagé l'informatisation totale c'est à dire depuis la conception initiale -dessin- jusqu'au prototype final (répondant aux critères et aux exigences du concepteur) d'un problème de résistance des matériaux déjà partiellement informatisé.

Nous avons créé une base SOMINE devant contenir les caractéristiques mécaniques des aciers. Cette base aura pour but d'aider le concepteur en lui permettant de construire un modèle (banque graphique) et d'obtenir une structure métallique ayant une tenue en service convenable et un prix de revient compatible avec l'usage à en faire. Pour cela il y a utilisation des caractéristiques mémorisées dans la base, celles-ci sont d'ailleurs obtenues par des calculs statistiques ou l'établissement de corrélations faites à partir de résultats expérimentaux.

La partie graphique n'a pas été expérimentée faute de matériel adapté au centre de calcul. Le calcul de la tenue en service d'une structure métallique a été exposé dans ce rapport. Du point de vue informatique cette étude a abouti à des améliorations et à un approfondissement des différentes possibilités de SOMINE ceci en vue de l'adapter au problème à résoudre. C'est dans cette partie que l'on trouve la caractéristique programme ainsi que la caractéristique procédure qui permettent de résoudre le problème des actions implicites dans la base. Il s'agit d'un travail d'informatique appliqué ou expérimentale et nous sommes conscients d'avoir mis en évidence plusieurs problèmes qui n'ont pas reçu de solution exploitable directement. Nous espérons cependant que les problèmes résolus contribueront à faire progresser la C.A.O., qu'ils seront exploités ou améliorés et qu'ils permettront de faire une synthèse des recherches déjà entreprises et des problèmes déjà résolus ou à résoudre dans ce domaine.

Ceci constitue le point d'une partie du travail qu'à réalisé notre équipe depuis deux ans que l'idée du produit SOMINE est née. Tant que les travaux continuent le système évolue et nous ne manquerons pas de publier ses perfectionnements et de faire le point de l'avancement des travaux dans chacun des domaines explorés. Nous espérons que notre équipe a la souplesse et l'adaptabilité de

notre produit mais nous souhaitons également posséder l'imagination et la créativité qui seules peuvent rendre notre système de bases de données utile et utilisé.

Nous pensons continuer de rechercher ou d'exploiter les voies d'utilisation ouvertes mais n'oublions pas qu'apprendre à créer et à utiliser une base SOMINE ne demande pas beaucoup de temps ni de peine. Nous espérons que cette facilité encouragera les utilisateurs éventuels que nous sommes prêts à renseigner et à former.

FICHE SIGNALETIQUE

**(extraite de "l'étude sur les SOFTWARES
de gestion et d'interrogation de Fichiers"
Délégation à l'Informatique, groupe
"Systèmes d'Information" -**

I - RENSEIGNEMENTS GENERAUX :

1.1. - Nom du système

SOMINE

1.2. - Réalisateurs :

Marcel GAILLARD

Pierre MARTY

Claudette SAYETTAT

Ecole Nationale Supérieure des Mines
SAINT - ETIENNE

1.3. - Personnes responsables:

les réalisateurs

II - CARACTERISTIQUES HARDWARES ET SOFTWARES :

2. 1. - Unité centrale

SOMINE est implémentable sur tout matériel.

2.2. - Taille mémoire centrale nécessaire

80 K octets

2.3. - Unités périphériques nécessaires :

- . lecteur de cartes
- . disques
- . imprimante
- . écrans alphanumériques ou graphiques

2.4. - Types de terminaux supportés

- . toute imprimante ou lecteur de cartes
- . tous écrans graphiques ou alphanumériques

2.5. - Nombre de terminaux supportés

limité par le temps de réponse seulement

2.6. - Taille mémoire centrale supplémentaire par terminal

non chiffrée actuellement

.../...

2.7. - Système d'exploitation utilisé

Système PHILIPS P 1000

2.8. - Langage de programmation du S.G.B.D.

SOMINE est programmé en FORTRAN (+ 50 instructions en assembleur PHILIPS)

2.9. - Taille du système

limitée par les matériels utilisés

2.10. - Processeurs nécessaires pour l'utilisation du système

- . Compilateur Fortran
- . éditeur de liens
- . système de gestion des terminaux

2.11. - Type du système

réseau

III - SOUPLESSE DU SYSTEME :

3.1. - Macrolangage de génération du système

inexistant

3.2. - Facilité d'utilisation du système

3.2.1. - Niveau requis par les utilisateurs du système

aucune compétence particulière

3.2.2. - Temps de formation de l'utilisateur

très rapide (2 ou 3 heures)

3.2.3. - Niveau requis par le responsable du système

niveau ingénieur

3.2.4. - Temps pour réaliser une application

Très variable suivant l'application. Il y a lieu de faire une analyse approfondie de la structuration des informations cela peut demander une dizaine de jours.

.../...

3.3. - Portabilité du système :

Peut être implémenté sur tout matériel ayant une taille mémoire suffisante.

3.4. - Comptabilité fichiers - langage de haut niveau

Accessibilité aux fichiers par langages de tous niveaux

IV - ORGANISATION DES FICHIERS :

4.1. - Structure logique :

4.1.1. - Types de structures logiques admis

- structures arborescentes à plusieurs niveaux
- structures de graphes (caractéristiques références)

4.1.2. - Organisation des fichiers :

4.1.2.1. - Dictionnaire des fichiers :

Les fichiers n'ont plus de signification logique, seules les adresses de mots sont prises en compte. Elle sont calculées par la gestion de mémoire.

4.1.2.2. - Descriptif d'un fichier :

Le fichier qui contient la structure est formé d'une suite de vecteurs de 15 mots mémoire contenant les renseignements nécessaires au calcul des adresses virtuelles des caractéristiques.

4.1.2.3. - Organisation des données :

A cause de la gestion de mémoire (transparente à l'utilisateur) la notion de fichier disparaît dans SOMINE. Les données sont caractérisées par leurs adresses virtuelles (dans l'espace virtuel) et c'est la gestion de mémoire qui calcule les adresses réelles des données sur le disque.

4.1.3. - Corrélations

4.1.3.1 - Au niveau des fichiers :

Les adresses dans le fichier des données sont calculées à partir des

.../...

éléments contenus dans le fichier structure.

4.1.3.2. - Au niveau enregistrement :

Dans SOMINE, la notion d'enregistrement n'a pas la signification habituelle. Il faudrait plutôt parler de réalisations d'entités définies par la structure des informations.

4.1.4. - Langage de description

L.D.S. défini par des automates

4.2. - Structure physiques

4.2.1. - Support physique des données :

Les données sont placées sur des disques magnétiques

4.2.2. - Organisation physique des données :

Dans la mémoire virtuelle, pour chaque entité, on réserve une place correspondant au nombre maximum de réalisations. Seules les zones de mémoire virtuelle qui contiennent de l'information sont projetées sur l'espace disque. Cette projection se fait par blocs de longueur variable (de 2^3 à 2^8 mots) appelés sous-pages.

4.2.3. - Méthode d'accès :

- . Accès séquentiel (limité à une seule réalisation d'entité)
- . Accès par les caractéristiques inverses ou par les expressions booléennes de telles caractéristiques.
- . Accès direct par le numéro de réalisation.

4.2.4. - Type d'adressage :

SOMINE calcule l'adresse réelle à partir de l'adresse virtuelle d'une caractéristique, il y a création de "squatters" (qui sont chaînés entre eux) lorsque plusieurs adresses virtuelles donnent la même adresse réelle. La gestion de ces "squatters" est assurée par la gestion de mémoire.

V - FONCTIONS REMPLIES PAR LE SYSTEME :

5.1. - Interface de création

5.1.1. - Nature du processeur de création

La création se fait en "batch" il y a analyse de la structure des informations

.../...

écrite en L.D.S., cette chaîne alphanumérique est ensuite transformée en vecteurs contenant les renseignements relatifs à chaque caractéristique et qui permettront de calculer son adresse virtuelle dans l'espace virtuel du fichier-données.

5.1.2. - Type de données admises :

- . chaînes alphanumériques de longueur inférieure à 1280 caractères
- . données numériques de type entier ou réel simple précision ou double précision.
- . liste de valeurs
- . chaîne d'inverse
- . pointeurs de référence
- . programmes, valeur + programme, procédure

5.1.3. - Possibilité de codage et de décodage

Les données sont stockées dans leur état d'utilisation

5.2. - Consultation :

5.2.1. - Nature du processeur de consultation

Programmes Fortran d'analyse et d'interprétation des chaînes alphanumériques que constituent les requêtes. L'action à entreprendre est déterminée par le mode de la requête : interrogation, mise à jour, création, suppression, fréquence.

5.2.2. - Mode de consultation :

SOMINE permet de mode conversationnel (écran,) et le mode "batch".

5.2.3. - Multi-accès

possibles

5.2.4. - Consultation multi-fichiers

Possible

5.2.5. - Bibliothèque de questions

Chaque utilisateur peut créer son propre recueil de questions dans chacun de ses programmes d'utilisation.

5.2.6. - Extraction des fichiers

Très simple puisque se sont des fichiers FORTRAN à accès direct.

.../...

5. 3. - Mise à jour :

5.3.1. - Nature du processeur de mise à jour :

Tout programme en conversationnel ou en "batch" écrit dans un langage algorithmique.

5.3.2. - Mode de mise à jour :

Dans la version actuelle de SOMINE les mises à jour se font à l'aide de requêtes à l'écran ou en "batch"

5.3.3. - Mise à jour au niveau des enregistrements

Cette mise à jour se fait sans problème mais le nombre maximum de réalisation d'une entité est limité à celui déclaré au moment de la définition de la structure.

5.3.4. - Mise à jour au niveau des informations élémentaires :

Se fait à l'aide de requêtes ponctuelles. Des précautions doivent être prises lorsque la base contient des caractéristiques programme (possibilité d'actions implicites)

5.3.5. - Mises à jour spéciales :

Il est possible de mettre à jour des caractéristiques filtrées par une caractéristique inverse ou une expression booléenne sur des caractéristiques inverses. Signalons également la possibilité d'effectuer des mises à jour groupées ou des mises à jour à valeur variable.

5.3.6. - Mise à jour de la structure logique des données

Semble nécessaire pour certains types d'applications des bases de données. Les travaux sont en cours.

5.4. - Possibilités de traitement :

SOMINE permet d'effectuer les requêtes de bases suivantes :

- interrogation
- mise à jour
- création
- suppression

- fréquence
- valeur

on peut facilement ajouter d'autres modes

5.5. - Editions :

5.5.1. - Nature du processeur d'édition

C'est l'utilisateur qui par une valeur donnée à une variable décide de la nature de l'édition :

- imprimante
- écran
- formatage (spécification FORTRAN)

5.5.2. - Support d'édition

- imprimante
- écran

5.5.3. - Tri

Peut être effectué par des instructions contenues dans le programme d'utilisation écrit en langage évolué.

VI - MAINTENANCE-SECURITE-PROTECTION-INTEGRITE

6.1. - Maintenance des fichiers :

6.1.1. - Réorganisation, rechargement des fichiers

Sauvegarde du système utilisé

6.1.2. - Statistiques d'occupation des fichiers

Non prévues dans le prototype

6.2. - Sécurité d'utilisation

6.2.1. - Possibilités de redémarrage du système

En cas de panne, les modifications effectuées dans les quatre pages (de 257 mots) qui se trouvent dans la mémoire centrale (plan primaire) sont perdues.

6.2.2 - Point de reprises

non

.../...

6.2.3. - Journal des manipulations effectuées sur les fichiers :
pourra être mis en oeuvre si des utilisateurs en sentent la nécessité.

6.3. - Protection :

6.3.1. - Identification de l'utilisateur :

Un code de passe peut précéder ou être **contenu** dans les requêtes et filtrer ainsi les entrées dans la base.

6.3.2. - Réglementation de l'accès à l'information

Un système analogue à l'identification de l'utilisateur peut être mis en oeuvre pour réglementer l'accès à l'information. SOMINE étant encore un prototype sa réalisation dépendra de la demande.

6.4. - Intégrité des fichiers :

Les références sont chaînées automatiquement mais c'est l'utilisateur qui s'assure de la cohérence de ses fichiers

VII - RENSEIGNEMENTS DIVERS :

7.1. - Encombrement des fichiers :

La notion de mémoire virtuelle entraîne un encombrement minimum des fichiers SOMINE

7.2. - Temps de réponse :

Variable suivant la nature du traitement. Dans le contexte actuel (multiprogrammation) il dépend de l'encombrement de la machine.

7.3. - Fiabilité :

Le produit est trop récent pour que nous donnions une réponse valable à cette question.

7.4. - Coût d'utilisation :

Par une bonne organisation du fichier-structure et une bonne formulation des requêtes, on doit **arriver** à une dépense de temps machine assez faible.

